TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Andreas Teder

Kõnekorpuste audiofailide automaatne eeltöötlus kõnesünteesi treenimiseks

Bakalaureusetöö (9 EAP)

Juhendaja: Liisa Rätsep, MSc

Tartu 2021

Kõnekorpuste audiofailide automaatne eeltöötlus kõnesünteesi treenimiseks

Lühikokkuvõte:

Kõnekorpustes võib esineda liigset vaikust, mis vähendab kõnekorpustel treenitud kõnesünteesi mudelite kvaliteeti. Käsitsi vaikuse eemaldamine on pikk protsess, kuid liiga lihtsad automaatsed meetodid võivad tekitada teisi defekte. Selle bakalaureusetöö eesmärk on luua erinevaid meetodeid kasutav programm, mis automaatselt eemaldaks kõnekorpustest vaikust ilma defekte loomata. Programmis on rakendatud energiat ja nullpunktimäära kasutav meetod ning akustilisi mudeleid kasutav meetod. Programmiga eeltöödeldakse kolme eestikeelset kõnekorpust, millest kahte kasutatakse kõnesünteesi mudelite treenimisel. Nii eeltöötluse kui ka kõnesünteesi mudelite kvaliteeti hinnatakse ja analüüsitakse. Selgus, et mõlemad meetodid parandavad kõnekorpuste kvaliteeti, kuid akustiline mudel annab üldjuhul parema tulemuse.

Võtmesõnad:

Heli, eeltöötlus, vaikuse tuvastamine, kõnesüntees

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Automatic Preprocessing of Speech Corpora's Audio Files for the Purpose of Speech Synthesis

Abstract:

Speech corpora may contain excessive silence, reducing the quality of speech synthesis models when used for training. Manually removing the silence is a long process, but using overly simple automatic methods may cause other defects. The purpose of this bachelor's thesis is to create a program, which utilizes different methods to automatically remove the silence from speech corpora without creating defects. The program utilizes a method based on energy and zero crossing rate along with a method based on acoustic models. The program is used to preprocess three Estonian speech corpora, two of which are used to train speech synthesis models. The quality of both the preprocessing and the speech synthesis models is graded and analyzed. It was concluded that both methods increase the quality of speech corpora, but the acoustic models give a better result overall.

Keywords:

Sound, preprocessing, silence detection, speech synthesis

CERCS: P170 Computer science, numerical analysis, systems, control

Sisukord

Sissejuhatus

Tänapäeval on erinevate eesmärkide jaoks kasutada mitmeid avalikke kõnekorpuseid, näiteks eestikeelne Common Voice[1]. Kõnekorpustes on helifailid, mis sisaldavad inimkõne, ning helifaile kirjeldavad failid, mis viivad kokku helifailide nimesid ja helifailis loetud tekste ehk transkriptsioone. Kuid selliste kõnekorpuste helifailides võivad esineda erinevad defektid. Sellised defektid on näiteks taustamüra, huulematsud ja liigne vaikus nii helifailide alguses, lõpus kui ka helifailide keskel. Defekte esineb eriti palju avalikult kogutud kõnekorpustes, kus kõnelejaid on tuhandeid, kuna kõnelejate mikrofonid ja salvestuskeskkonnad ei pruugi olla sama kvaliteetsed. Defektide eemaldamiseks on vaja kõnekorpustele teha eeltöötlust.

Keskendudes liigse vaikuse eemaldamisele, on üks võimalikest lahendustest määrata helifailis oleva kõne alguse ja lõpu hetk ning hiljem arvestada ainult nende kahe hetke vahelist aega. Sellise lähenemisega saab vähendada helifaili ääres olevat vaikust üksikute ütluste korral [1] ning ka mehhaanilise müra olemasolul [2]. Kui helifailis loetakse ette mitu ütlust, mille vahel on pikad vaikused, ei piisa kogu kõne alguse ja lõpu hetke märkimisest, sest ütluste vaheline vaikus säilib. Et ütlustevahelist vaikust eemaldada, on kasutusel tükeldamise ja liigitamise lähenemine, mida kasutasid ka B. Chandu jt [3] linnuhäälte analüüsimiseks. Sellise lähenemise korral jagatakse helifail tükkideks ning iga tüki korral otsustatakse, kas tegu on vaikusega või kõnega.

Vaikuse eemaldamisega saavutatakse mitmeid positiivseid efekte. Näiteks on töödeldud helifailid vaikuste arvelt lühemad kui originaalid, seega väheneb kõnekorpuste maht, kuid kõne ise säilib. Samuti muutub antud helifailide edasine töötlus kiiremaks, kuna vaiksele osale ei pea enam töötlust rakendama. Lisaks on L. Rätsep jt [4] oletanud, et liigne vaikus kõnekorpustes võib vähendada nendel kõnekorpustel treenitud kõnesünteesi mudelite täpsust. Samas oletati, et liiga agressiivne vaikuse eemaldamine võib mudelite täpsust vähendada, mis teeb vaikuse eemaldamise keerulisemaks.

Käesoleva töö eesmärk on luua programm, mis automaatselt parandaks eestikeelsete kõnekorpuste audiofailide kvaliteeti erinevate eeltöötluste meetoditega. Lisaks analüüsitakse loodud eeltöötluse mõju nii audiofailide kvaliteedile kui ka eeltöödeldud audiofailidest loodud kõnesünteesi mudelite täpsusele. Täpsemalt keskendub töö audiofailides olevate varieeruva pikkusega vaikuste äratundmisele, eemaldamisele ja kärpimisele. Töö käigus loodud programmi rakendatakse varasemalt tehtud närvivõrgupõhise kõnesünteesi arendamise projekti [4] kõnekorpustele. Seega luuakse töö käigus ka parandatud kõnekorpused, millel treenitud kõnesünteesi mudeleid saab võrrelda projektis varasemalt loodud mudelitega[2]. Kui uued mudelid on vanadest täpsemad, siis saab uusi mudeleid ja parandatud kõnekorpuseid projektis rakendada.

Töö 1. peatükis selgitatakse helitöötluse tehnikaid ning tutvustatakse kõnesünteesi. 2. peatükis kirjeldatakse kasutatud kõnekorpuseid ja selgitatakse kõnekorpuste eeltöötluse vajadust täpsemalt. 3. peatükis on kirjeldatud loodud eeltöötluse meetodite tööpõhimõtteid ning 4. peatükis on välja toodud eelnevate meetodite ja loodud mudelite tulemused ning tulemuste analüüs.

[1] https://commonvoice.mozilla.org/et/datasets
[2] https://github.com/TartuNLP/deepvoice3_pytorch/releases/tag/kratt-v1.0

1. Helitöötluse ja kõnesünteesi teoreetiline taust

Järgnevalt antakse ülevaade erinevatest helitöötluse tehnikatest, kõnesünteesist ning kõnekorpuste loomise protsessist.

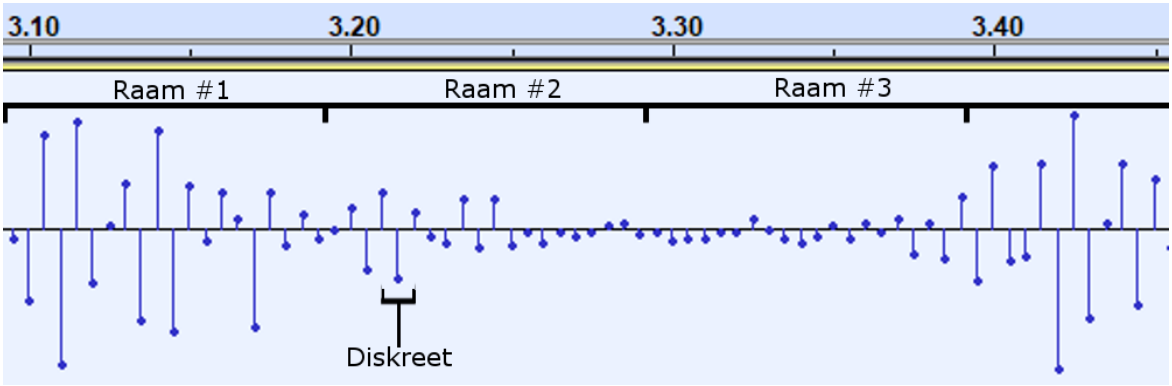
1.1 Ülevaade helitöötluse tehnikatest

Helist vaikuse või kõne alguse ja lõpu ära tundmiseks on mitmeid meetodeid, mille kasulikkus sõltub palju töödeldava heli omadustest. Järgnevalt kirjeldatakse erinevaid tunnuseid, mille abil saab helis olevat vaikust või kõnet ära tunda, ning tunnuste leidmist abistavaid transformatsioone.

1.1.1 Raamistamine

Heli töötlemisel tegeletakse tavaliselt digitaalse helisignaaliga, mis koosneb ühtlaste vahedega diskreetidest (ingl *sample*), mis kirjeldavad heli amplituudi mingil ajahetkel. B. Chandu jt [3] kirjutasid, et kuna helis on palju ajas varieeruvaid tunnuseid, tuleb enne töötlemist jagada helisignaal raamideks (ingl *frame*), kus on kindel hulk diskreete. Raami suuruse valimisel arvestasid nad nii diskreetimissageduse (ingl *sampling rate*) kui ka heli kogupikkusega. G. S. Ying jt [5] näitasid, et heli kogupikkusega ei pea alati arvestama ning valisid raami suuruseks 250 diskreeti, mis nende diskreetimissageduse juures kestab 20 ms. Kui tahetakse liigitada, mis ajahetkel on helis tegu vaikusega, aitab raamide liigitamine vaikuste ajahetke paremini määrata kui kogu heli liigitamine, sest kogu heli vaikuseks liigitamine eemaldaks ka helis oleva kõne.

Joonisel 1 on raamistamise näiteks kujutatud Tartu Ülikooli (edaspidi UT) uudislausete kõnekorpusest [6] pärit helifaili helisignaali, mille diskreetimissageduseks on 200 Hz, seega iga sekundi kohta on signaalis 200 diskreeti. Raami suuruseks on valitud 20 diskreeti. Reaalselt on diskreetimissagedus ja raami suurus suuremad, näiteks 12.5 kHz ja 250 diskreeti vastavalt [5]. Lisaks võivad raamid kattuda (ingl *overlap*) [5], seega üks diskreet võib olla korraga mitmes raamis.



Joonis 1. Diskreetimissagedusega 200 Hz helisignaali märgendatud kujutis. X-teljel on helisignaali aeg ja Y-teljel amplituud.

Raamistamine on helitöötluses vajalik samm, mis võimaldab hinnata ajas muutuvaid tunnuseid helis, näiteks helitugevust. Lisaks võimaldavad raamid liigitada terve heli asemel väikseid tükikesi, mis aitab eemaldada helist vaikust sisaldavaid osasid.

1.1.2 Akna funktsioon

Kuigi raamid võimaldavad heli tükkideks jagada, ei muuda raamid nendes olevate diskreetide amplituudi. Et raamide ääri sujuvamaks muuta, kasutatakse akna funktsiooni. National Instruments on loonud ülevaate „*Understanding FFTs and Windowing*“ [7], mille järgi

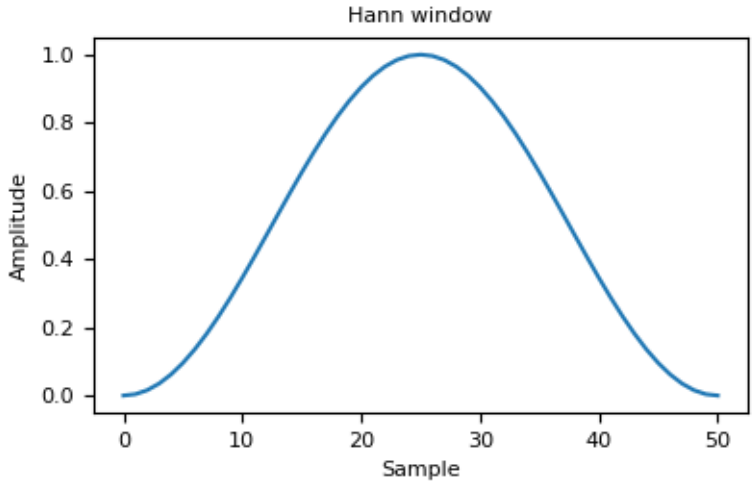
nimetatakse aknastamiseks (ingl *windowing*) ehk akna funktsiooni rakendamiseks protsessi, mis vähendab lõplike jadade otspunktides olevate diskreetide amplituude. Selle jaoks korrutatakse jada aknaga, mis sujuvalt liigutab jada otspunktide amplituude nulli poole. Kuna raamid on jadad, mis koosnevad lõplikul hulgal diskreetidest, saab aknastamist rakendada ka raamidele.

Raamide aknastamisel saadakse tulemuseks raamid, kus raami keskel olevad diskreedid säilitavad üldiselt oma amplituudi, kuid raamiäärsete diskreetide amplituud läheneb nullile. Sellist efekti saab kasutada, et tõsta raami keskel olevate diskreetide mõju helis olevate tunnuste otsimisel või vähendada raamide kattuvust.

Kuigi aknafunktsioone on mitmeid, sobib enamikul juhtudel Hanni aken [7], mida kutsutakse ka Hanning aknaks. Pythoni paketis NumPy kasutatud Hanni akna valem on toodud välja valemis 1 [8], kus *M* on diskreetide arv väljundi aknas.

$$w(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{M-1}\right) \qquad 0 \le n \le M-1 \ (1)$$

Joonisel 2 on näha Hanni akent, kui diskreetide arv väljundi aknas on 51. Joonisel on näha Hanni akna kõrget väärtust keskpunktis olevate diskreetide korral ja sujuvat lähenemist nullile, mida kaugemale akna keskpunktist minnakse. Sellise akna korrutamisel 51 diskreeti suure raamiga säiliksid Hanni akna omadused raamis.



Joonis 2. Hanni aken, kus X-teljel on diskreedi number ja Y-teljel amplituud [8].

Aknastamine on helitöötluses raamimisele järgnev etapp, mis teeb raamide diskreetide jaotuse naturaalsemaks. Lisaks muudab aknastamine raamid sümmeetrilisemaks, kuna raamide otspunktide amplituudid koonduvad nulli. See on kasulik omadus, mida rakendatakse keerulisemate heli kujutiste juures, näiteks spektrogrammi luues.

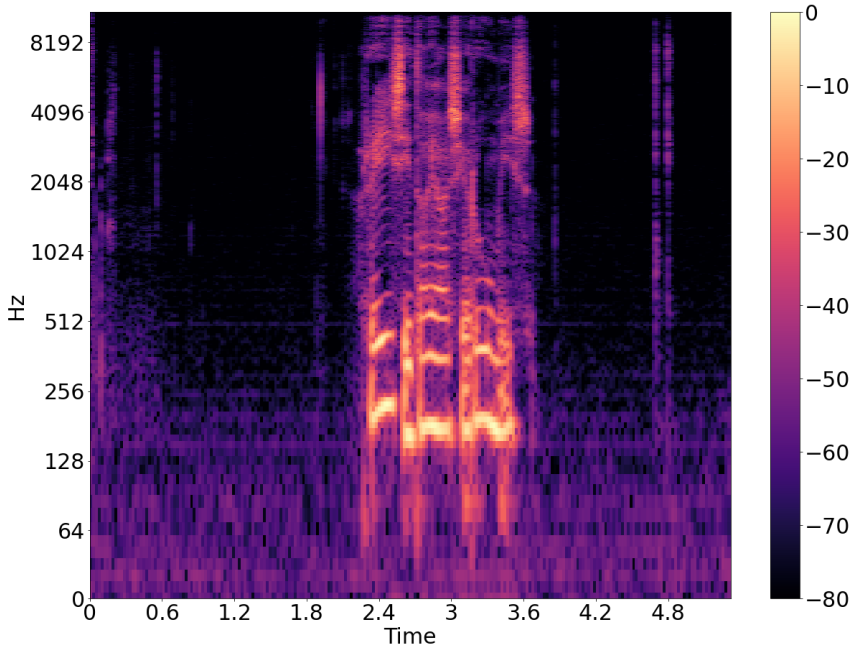
1.1.3 Spektrogramm

Tavaliselt on kõnekorpuses olevad helifailid ajadomeenis, mille puhul saab diskreetide väärtusi vaadates teada, mis oli heli amplituud mingil kindlal ajahetkel. Samas ei anna ajadomeen informatsiooni selle kohta, mis sagedusega heli on helifailis dominantne. Selline informatsioon võib olla kasulik, kuna näiteks inimhääle ja müra sagedus võib olla erinev. Samas ei anna kogu heli kohta ühekordne sageduse leidmine palju informatsiooni, sest rääkides muudab inimene pidevalt oma hääle sagedust ja kogu helifaili liigitamine ei aita

helifaili sees muudatusi teha. Probleemi saaks lahendada, kui helisignaali raamistada ja iga raami kohta leida heli sagedus, nõnda säilib informatsioon nii aja kui ka sageduse kohta.

Eeltoodud probleemi lahenduseks on näiteks lühiaja Fourier' pööre (ingl *short-time Fourier transform* ehk STFT). V. Velardo [9] kirjelduse järgi tuleb STFT rakenduseks esiteks helisignaal raamideks ja akendeks jagada, kus aknaks nimetatakse raami, millele on rakendatud aknafunktsiooni. Aknastamisega saab vähendada Fourier' pöörde kasutamisel tekkivat spektri leket (ingl *spectral leakage*), mis vähendab Fourier' pöörde täpsust [7, 10]. Pärast saab igale aknale rakendada diskreetset Fourier' pööret (ingl *discrete Fourier transform* ehk DFT), et aken jagada sageduskomponentideks. Sageduskomponentide arv on DFT puhul piiratud, seega saab teada ainult kindla hulga lineaarsete vahedega sageduste magnituudi, näiteks sagedused 2 Hz, 4 Hz, 6 Hz jne. Erinevalt amplituudist on magnituud absoluutväärtus, mis STFT puhul näitab mingi sageduse esinemise tugevust. V. Velardo järgi saab akendest ja nende sageduskomponentidest luua spektrogrammi, kus on kajastatud nii aeg, sagedus kui ka magnituudi ruut. Töös defineeritakse spektrogramm J. O. Smithi [11] näitel, kellest lähtuvalt on spektrogramm *STFT magnituudi intensiivsuse kujutis logaritmilisel skaalal, näiteks dB.*

Joonisel 3 on STFT illustreerimiseks kujutatud UT kõnekorpuse [6] helifaili spektrogramm, milles loetakse sisse lause „Reisid ajas ja ruumis." Spektrogrammi keskel olev inimhääle osa erineb vaikusest märgatavalt. Inimhääle osa juures on sageduskomponentide magnituud kõrgem ning reeglipärasem. Vaikuse puhul on sageduskomponentide magnituudid ühtlasemalt jaotunud, kuid inimhääle puhul on märgata domineerivaid sagedusi. Samas on näha ka spektrogrammi alguses ja lõpus ajahetki, kus hiireklikkide tõttu on heli võimsam.

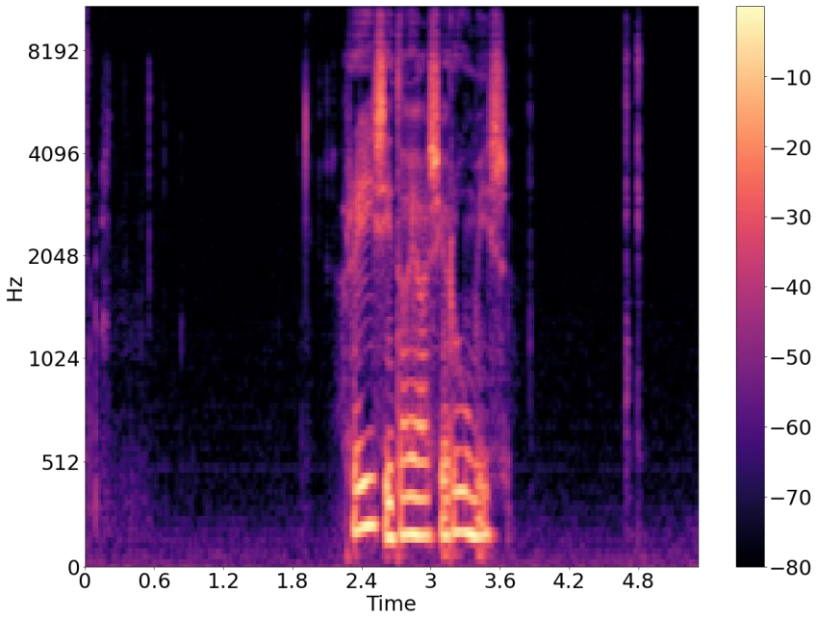


Joonis 3. Lause „Reisid ajas ja ruumis," STFT abil loodud spektrogramm, kus X-teljel on aeg, Y-teljel sagedus ja värv näitab magnituudi ruutu detsibellides (skaala on nihutatud).

Peale STFT abil loodud spektrogrammi, kasutatakse ka mel-spektrogrammi. V. Velardo [12] kirjelduse järgi *tunnetavad inimesed heli logaritmiliselt*. Näiteks kuulevad inimesed suurt erinevust 50 Hz ja 250 Hz sagedusega helil, kuid väiksemat erinevust 1500 Hz ja 1700 Hz sagedusega helil, kuigi erinevus on mõlemal juhul 200 Hz. STFT abil loodud spektrogramm ei arvesta inimese perspektiiviga, vaid sageduskomponendid on valitud lineaarsete vahedega. V. Velardo järgi saadakse mel-spektrogramm STFT abil loodud

spektrogrammist, kus originaalse spektrogrammi sagedused jagatakse logaritmiliselt ümber ning tulemuseks on mel-spektrogramm, kus sageduskomponentide vahed on võrdsed inimese tunnetuse perspektiivist. Inimese perspektiivist spektrogrammi kasutatakse närvivõrkude [13] juures, et ennustada, kuidas mingi lause mel-spektrogramm välja näeb ning hiljem muudetakse ennustatud mel-spektrogramm heliks tagasi.

Joonisel 4 on mel-spektrogrammi illustreerimiseks kujutatud UT kõnekorpuse [6] helifaili mel-spektrogramm, milles loetakse sisse lause „Reisid ajas ja ruumis.“ Kuna mel-spektrogrammi eesmärk on sarnane STFT abil loodud spektrogrammile, siis ei erine joonised 3 ja 4 palju, kuid kuna mel-spektrogrammi puhul kasvab sageduskomponentide vahe eksponentsiaalselt, on joonis 3 kõrgemate sageduste puhul täpsem.



Joonis 4. Lause „Reisid ajas ja ruumis,“ mel-spektrogramm, kus X-teljel on aeg, Y-teljel on sagedus ja värv näitab magnituudi ruutu detsibellides (skaala on nihutatud).

Spektrogramm on kasulik tööriist, mille abil saab heli visualiseerida. Peale ülevaate andmise saab spektrogrammi loomiseks kasutatud magnituude rakendada tunnustena, et eristada erinevaid helisid. Kuid helist tunnuste leidmiseks on kasutusel ka teisi meetodeid.

1.1.4 *Root mean square energy* ja *short-time energy*

Helitöötluse juures on ajadomeenis vaikuse ära tundmiseks populaarne tunnus ruutkeskmine energia (ingl *root mean square energy* ehk RMSE). G. S. Ying jt [5] defineerisid RMSE kui *ruutjuure signaali diskreetide amplituudide ruutude summa aritmeetilisest keskmisest*, ehk eeldades, et helisignaal on jagatud suurusega W raamideks, $s_n(i)$ on raami number n diskreedi number i väärtus ja E_n on raami number n RMSE, defineeritakse E_n valemis 2 näidatud viisil.

$$E_n = \left[\frac{1}{W}\sum_{i=1}^{W} s_n^2(i)\right]^{\frac{1}{2}} \tag{2}$$

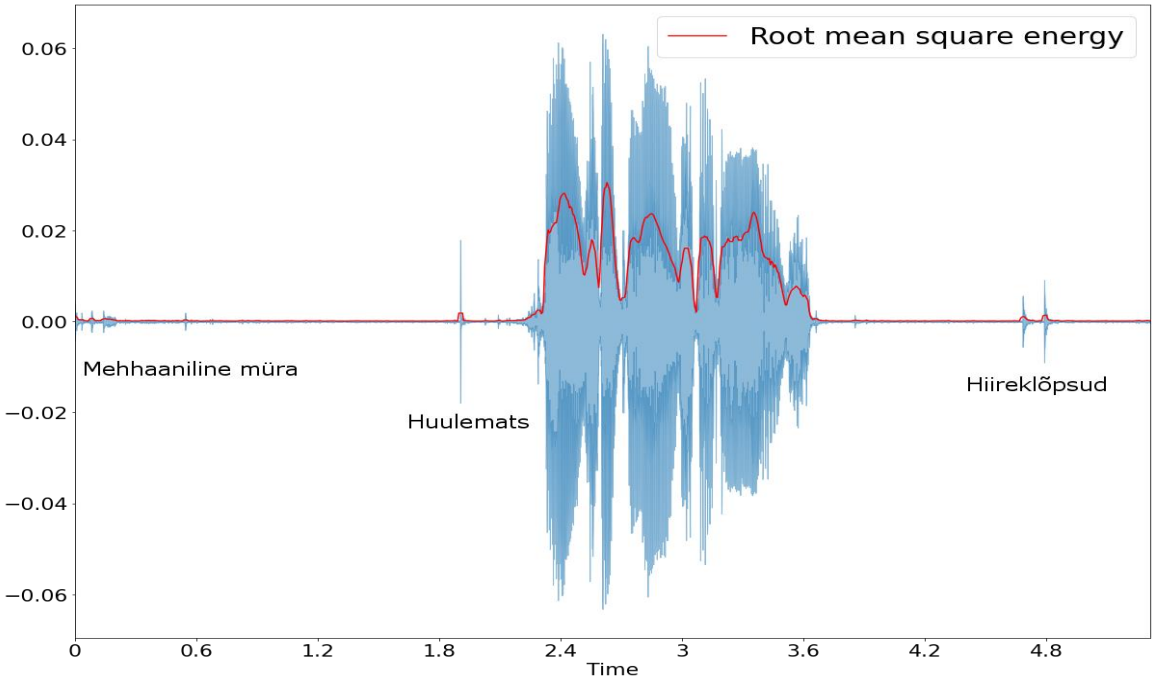
Sarnase eesmärgiga tunnus on lühiaja energia (ingl *short-time energy)*. T. H. Zaw ja N. War [14] defineerisid lühiaja energia valemis 3 näidatud viisil, kus N on raami suurus, $x(i)$ on originaalne signaal ja $E(n)$ on raami number n lühiaja energia. Lühiaja energia pole ühtselt defineeritud, sest lühiaja energia definitsioonis on varasemalt rakendatud ka

aknafunktsiooni [15] ning võetud amplituudist ruudu asemel absoluutväärtus [1]. Siiski ei mõjuta need muudatused lühiaja energia põhilist eesmärki - mõõta helisignaali energiat.

$$E(n) = \sum_{i=1}^{N} x_n^2(i) \tag{3}$$

Energiapõhised tunnused annavad hea alguse helis kõne märgendamisele, kuid varasemates uuringutes on kombineeritud see tunnus tavaliselt millegi muuga. Näiteks kombineerisid L. R. Rabiner ja M. R. Sambur [1] lühiajaenergia nullpunktimääraga (ingl *zero crossing rate*), et täpsustada lühiajaenergia märgendamist, kui kõnes alguses või lõpus esineb helitu kõne, näiteks helitu häälik. Samas leidsid A. Ganapathiraju jt [16], et energiapõhise tunnuse kombineerimine nullpunktimääraga ei tõsta märgendamise täpsust märgatavalt. Nullpunktimäär pole ainus tunnus, mida saab lühiajaenergiaga kooskõlas kasutada. L.-S. Huang ja C.-H. Yang [2] kombineerisid lühiajaenergia entroopiaga, kuna ainult lühiajaenergiat kasutades langeb täpsus mehhaanilise müra, näiteks auto mootori müra, olemasolul.

Joonisel 5 on RMSE illustreerimiseks kujutatud UT kõnekorpuse [6] helifail, milles loetakse sisse lause „Reisid ajas ja ruumis.“ Kasutatud on raami suurusega 512 diskreeti ja helisignaali diskreetimissageduseks on 22 050 Hz, seega ühe raami kestvus on umbes 23 ms. Joonisel on näha, et RMSE väärtus on kõrge seal, kus esineb kõne, kuid tuleks tähele panna, et helisignaali lõpus olevad kaks valjemat heli ei ole inimkõne, vaid hiireklõpsud, mille ajal on RMSE väärtus samuti tavalisest kõrgem. Sama kehtib enne kõne esineva huulematsu ja signaali alguses oleva mehhaanilise müra kohta.



Joonis 5. Lause „Reisid ajas ja ruumis,“ helisignaal koos ruutkeskmise energiaga. Raami suurus on 512 diskreeti ja diskreetimissagedus on 22 050 Hz. X-teljel on aeg ja Y-teljel nii signaali amplituud kui ka RMSE väärtus.

Kuna RMSE on energiapõhine, on see hea tunnus, mille põhjal vaikuse leidmist alustada. Piisavalt müravabas keskkonnas on RMSE erinevus kõne ja vaikuse vahel märgatav, kuid helitute häälikute täpsustamiseks on vaja RMSE kombineerida teiste tunnustega.

1.1.5 *Zero crossing rate*

Peale energia on ajadomeenis kõne ära tundmiseks kasulik tunnus nullpunktimäär (ingl *zero crossing rate* ehk ZCR). T. H. Zaw ja N. War [14] kirjeldasid nullpunktimäära kui *nullpunktide esinemiste arvu sekundis*. Nullpunkt tähendab nende järgi juhtu, kus järjestikused diskreedid on erineva märgiga. ZCR on defineeritud valemis 4, kus *s*(*m*) on originaalne signaal ja Z_n on ZCR.

$$Z_n = \sum_{m=-\infty}^{\infty} |\mathrm{sgn}[s(m)] - \mathrm{sgn}[s(m-1)]| \tag{4}$$

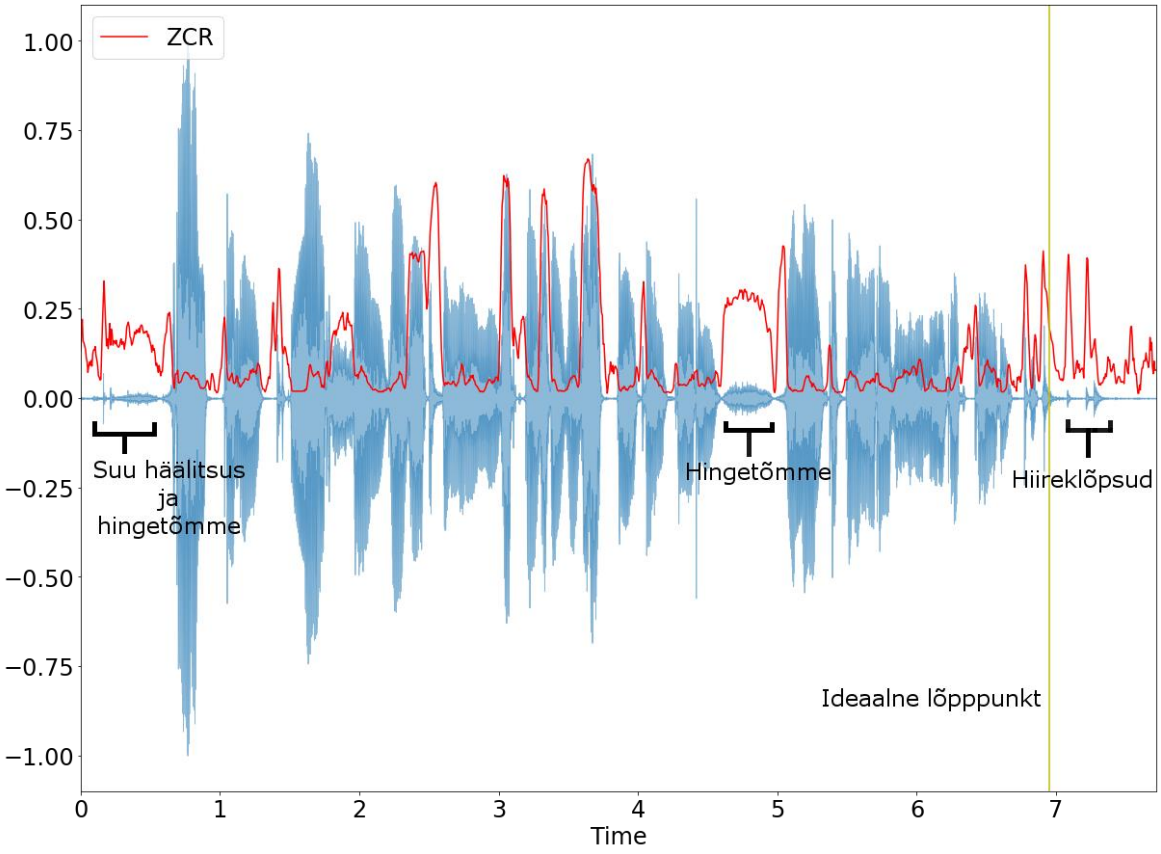
Valemis 4 kasutatud funktsioon sgn[] on defineeritud järgnevalt:

$$\mathrm{sgn}[x] = \begin{cases} 1, (x \geq 0) \\ -1, (x < 0) \end{cases}$$

T. H. Zaw ja N. War on kirjutanud, et ZCR väärtus on suurem kõrge sagedusega signaalides ning kõne puhul viitab suur ZCR helitule kõnele. Võib oletada, et helitu kõne koosneb eesti keeles helitutest häälikutest, nagu g, b, d ja f, mille korral signaali energia on madal.

ZCR väärtust ei pea leidma kogu signaali kohta, nagu on toodud valemis 4, vaid võib signaali enne raamistada ning seejärel leida iga raami ZCR väärtuse. Sellisel juhul saab iga raami kohta teada, kas seal asub helitu häälik või mitte.

Joonisel 6 on nullpunktimäära illustreerimiseks kujutatud kõnekorpuse [6] helifail, milles loetakse sisse lause „Raudteelõik Nordrhein Westfalenis asuva Siegburgi ja Montabauri vahel on blokeeritud.“ Kasutatud on RMSE näitega identset raami suurust ja diskreetimissagedust. ZCR väärtuse skaala on nihutatud näitama, kui suur osa raami diskreetidest põhjustasid nullpunkti. Joonisel on märgendatud kollaselt ideaalne lõpppunkt ning on näha, et sellel ajahetkel on ZCR väärtus veel kõrge, kuid signaali amplituudi väärtus võrdlemisi madal. Väärib mainimist, et loetud lause lõppeb helitu häälikuga, mille korral on tavaliselt ZCR väärtus kõrge. Joonisel on märgata ZCR väärtuste tõusu enne ja pärast kõne, kuid erinevate häälitsuste ajal, nagu hingetõmbed, tõuseb ZCR väärtus samuti märgatavalt.



Joonis 6. Lause „Raudteelõik Nordrhein Westfalenis asuva Siegburgi ja Montabauri vahel on blokeeritud," helisignaal koos nullpunktimääraga. Raami suurus on 512 diskreeti ja diskreetimissagedus on 22 050 Hz. X-teljel on aeg ja Y-teljel nii signaali amplituud kui ka ZCR väärtus.

Kuna ZCR väärtus helilise kõne ja vaikuse puhul on piisavalt sarnane, siis ainult ZCR leidmine ei aita helis vaikust täpselt leida. Kuid ZCR helitute häälikute äratundmise omadust saab kasutada kooskõlas energiapõhiste tunnustega, et täpsustada leitud kõne vahemikke [1].

1.2 Ülevaade kõnesünteesist ja kõnekorpustest

Kõnesünteesi eesmärk on digitaalset teksti teha inimkõneks. Selliseid süsteeme kasutatakse pimedatele või lugemisraskustega inimestele mõeldud programmides, näiteks veebilehtede ettelugemiseks, tehisintellekti kasutavates suhtlustarkvarades, nagu Siri, või uue keele õppimises, et kuulata võõrast keelt väliskeskkonnas viibimata.

Kõnesünteesi mudelite loomiseks on mitmeid meetodeid. Näiteks on EKI loonud Markovi peitmudelil põhinevaid kõnesünteesi mudeleid[3]. Peale selle on populaarsed tehisnärvivõrgud, mis leiavad kasutust ka süsteemis Deep Voice 3, mida kasutatakse teadustööga seotud kõnesünteesi projektis.

W. Ping jt [13] loodud Deep Voice 3 on täielikult konvolutsiooniline tehisnärvivõrgupõhine kõnesünteesi süsteem. Süsteemi sisendiks on tekst ning väljundiks mel-spektrogramm, mis konverteeritakse ajadomeenis olevaks signaaliks. Autorid on kirjutanud, et süsteem võib olla mitmehäälne, vähendab sünteesitud kõne vigu, nagu sõnade kordamine või vahele jätmine, ning võrreldes teiste samal ajal loodud süsteemidega on Deep Voice 3 kiirem.

[3] https://www.eki.ee/heli/index.php/k%C3%B5nes%C3%BCntees

Teadustööga seotud projektis on kasutusel R. Yamamoto Deep Voice 3 PyTorch implementatsioon[4], mis on muudetud eesti keelt toetama[5]. Seega kasutatakse ka teadustöö raames kõnesünteesi mudelite loomiseks Deep Voice 3 süsteemi.

Kuna Deep Voice 3 süsteemi hindamiseks pidi mudeleid treenima, tegelesid W. Ping jt [13] muuhulgas ka helifailides oleva vaikusega. Vaikuse ja pauside eemaldamise asemel muutsid nad helifailide transkriptsioone, märkides transkriptsioonis olevate sõnade vahele pauside pikkused. Märgenditena kasutati nelja erinevat sümbolit, mille kasutus sõltus pausi kestvusest. Deep Voice 3 loomisel märgendati transkriptsioone nii käsitsi kui ka automaatse tekstijoondaja Gentle[6] abil. Kuigi W. Ping jt pole välja toonud, kuidas tegeleti pikkade vaikustega helifailide otspunktides, näib ühe süsteemis kasutatud kõnekorpuse, LibriSpeech [17], kuulamisel, et kasutatud kõnekorpustes on helifailide algusest ja lõpust pikad vaikused juba eelnevalt eemaldatud.

Kõnekorpuste loomist saab samuti teha nii käsitsi kui ka automatiseeritumalt [17, 18]. Kuna uute kõnekorpuste lindistamine on aeganõudev protsess, on populaarne lähenemine olemasolevate keeleressursside, näiteks audioraamatute, kasutus. Audioraamatute kasutuse puhul on peatükkide kaupa helifailid ja transkriptsioonid olemas, kuid kõnekorpuse loomiseks peab audiofailid ja transkriptsioonid osadeks tükeldama ning omavahel kokku viima. Lisaks võiks transkriptsioone muuta häälduspärasemaks, sest kõnekorpuseid kasutatakse nii kõnesünteesis kui ka kõnetuvastuses, kus numbrid ning lühendid vähendavad mudelite kvaliteeti.

Automaatset lähenemist kasutasid V. Panayotov jt [17], kes on LibriSpeech kõnekorpuse loojad. Kõnekorpus põhineb LibriVox[7] audioraamatutel, millest loodi 1000 tunni pikkune ingliskeelne kõnekorpus. Et tükeldada 1000 tundi kõne ja viia tükid vastavusse transkriptsioonidega, eeltöödeldi transkriptsioonid ja rakendati helifailidele kõnetuvastust, et saada uued tuvastatud transkriptsioonid. Kõnetuvastust tehes saadi samuti teada, mis ajahetkel esines helifailis mingi sõna või häälik. Leides tekstiosi, kus tuvastatud transkriptsioon sarnaneb tugevalt originaalsele, ja teades sõnade asukohti helifailis, saadi kindlamate tekstiosade juures helifaile ja transkriptsioone pauside kohalt tükeldada. Loodud tükkide puhul kontrolliti häälikute põhjal, kas helifailis loetu vastab transkriptsioonile ning vigased tükid eemaldati. Kuna sobivate tükkide pikkus oli kuni 35 sekundit, korrati tükeldamist, et helifailide kestvus oleks sobilikum.

K. Park ja T. Mulc [18] kasutasid samuti LibriVox audioraamatuid, kuid nende eesmärk oli kõnekorpuste loomine teistele keeltele peale inglise keele. Kuna teiste keelte kohta pole nii palju andmeid, on nende loodud kõnekorpuste[8] kogumaht alla 150 tunni. Erinevalt V. Panayotovi jt lähenemisest, proovisid kõnekorpuse loojad tekstijoondaja Gentle ning helitöötlusprogrammi Audacity abil helifaile tükeldada ja iga tüki vastava transkriptsiooniga ühendada. K. Park ja T. Mulc kirjutasid, et hiljem otsustati lasta ekspertidel käsitsi igale tükile joondamist teha, sest Gentle oli liiga keelespetsiifiline ning ebatäpne. Autorite kogemusest võib oletada, et kui kasutada eestikeelsete kõnekorpuste vaikuse eemaldamiseks tekstijoondajaid või treenitavaid mudeleid, peaksid need olema eesti keele spetsiifilised.

[4] https://github.com/r9y9/deepvoice3_pytorch
[5] https://github.com/TartuNLP/deepvoice3_pytorch
[6] https://github.com/lowerquality/gentle
[7] https://librivox.org/
[8] https://github.com/Kyubyong/CSS10

2. Kõnekorpused

Kasutatavaid kõnekorpused ei ole ainult võõrkeeltes. Töö raames kasutatakse EKI ilukirjanduse [19] ja UT uudiste kõnekorpust [6] eeltöötluse arendamiseks ja kõnesünteesi mudelite loomiseks. Lisaks kasutatakse ERR uudiste kõnekorpust [20], et kontrollida eeltöötluse efektiivsust uues keskkonnas. Järgnevalt põhjendatakse kõnekorpuste eeltöötluse vajadust ja antakse ülevaade kasutatud kõnekorpustest koos nende unikaalsete omadustega.

2.1 Kõnekorpuse automaatse eeltöötluse vajadus

Kõnekorpuses esinevatel helifailidel võib esineda probleeme, mis vähendavad nende põhjal treenitud kõnesünteesi mudelite kvaliteeti. L. Rätsep jt [4], kes tegelevad eestikeelse kõnesünteesi arendamise projektiga, hindasid ja võrdlesid nii projekti raames loodud Deep Voice 3 tehisnärvivõrke kasutavaid mudeleid, varasemalt loodud Eesti Keele Instituudi Markovi peitmudelil põhinevaid mudeleid kui ka Google'i[9] kõnesünteesi. Projekti raames hinnati peale üldise kvaliteedi ka kõnesünteesi mudelite spetsiifilisi probleeme. Selleks paluti hindajatel iga mudeli korral kuulata erinevaid sünteesitud helifaile ning üles märkida kõnesünteesi mudelil esinevaid probleeme, sealhulgas sõnade vahele jätmine, poolikud laused, liiga järsult lõppevad ja algavad laused ning helitugevuse probleemid. Samuti paluti hinnata originaalseid helifaile, mis pärinesid treenimiseks kasutatud kõnekorpustest. Hindamise tulemusel selgus, et loodud Deep Voice 3 mudelitel on teiste mudelitega võrreldes rohkem probleeme järsu lauselõpu ja -algusega, helitugevusega ning sõnade vahele jätmisega, lisaks esineb mõnes Deep Voice 3 mudelis poolikuid lauseid. Treenimiseks kasutatud kõnekorpustes esines eelnevatest probleemidest vaid järsk lauselõpp ja -algus, kuid mitte nii tihti kui loodud kõnesünteesi mudelites.

L. Rätsep jt kirjutasid, et kindlate treenimiseks kasutatud kõnekorpuste korral esines varieeruva pikkusega vaikus helifailide algul, mis võis põhjustada treenitud kõnesünteesi mudelites sõnade vahele jätmist ja samuti esines mõnes kõnekorpuses liiga agressiivne vaikuse eemaldamine helifaili lõpust, mis võis põhjustada järske lauselõppe treenitud kõnesünteesi mudelites. Lisaks autorite järeldustele võiks oletada, et kõnekorpuses esinevad järsu lauselõpu ja -alguse probleemid võimendatakse loodud kõnesünteesi mudelites, kuna korpustes endis oli võrreldes loodud mudelitega neid probleeme vähem. Seega saab oletada, et kõnesünteesi mudelite kvaliteeti saab tõsta, kui parandada vaikuse eemaldamise eeltöötlust kõnekorpustele.

L. Rätsepaga suhtlusest selgus, et eelmises lõigus kirjeldatud projekti arendati edasi pärast hindamise tulemusi. Täpsemalt rakendati projektis kasutatud kõnekorpustele eeltöötlust helitöötlusprogrammiga Audacity[10], kuid tulemuseks saadud helifailides esines ikkagi liiga tihti järsk lõpp. Kasutatud Audacity meetodite hulgas oli heli normaliseerimine, lausetesiseste vaikuse kärpimine ja vaikuse eemaldamine otspunktidest. Sellest saab järeldada, et vaikuse eemaldamiseks kõnekorpusest on vaja midagi enamat kui staatiliste meetodite kogumit, mille parameetrid jäävad samaks sõltumata helifaili sisust, või tuleks sellisele meetodite kogumikule lisada mõni uus meetod. Mõlemal juhul ei piisa lihtsast eeltöötlusest, vaid peab mõtlema teiste lähenemiste peale.

Peale kvaliteedi probleemi, tekkis Audacity kasutamisel projektis probleem automatiseerimisega, kuna helifailide töötlus programmiga võttis palju aega ja Audacity pole kergesti lisatav kõnesünteesi töövoogu, nagu mõni Pythoni pakett. Kuna enamasti kasutatakse suurte

[9] https://translate.google.com/
[10] https://www.audacityteam.org/

kõnekorpuste töötluseks ja kõnesünteesi mudelite treenimiseks arvutusklastreid, kus valdav tööriist on käsuviip, siis graafilisele liidesele keskenduv Audacity ei pruugi olla kõige efektiivsem valik. Sellel põhjusel võttis protsess ka palju aega, sest Audacity ei töötanud võimsas arvutusklastris, vaid isiklikes sülearvutites. Seega kiiruse ja mugavuse huvides võiks eeltöötlus olla kergesti automatiseeritav.

2.2 Eesti Keele Instituudi ilukirjanduse kõnekorpus

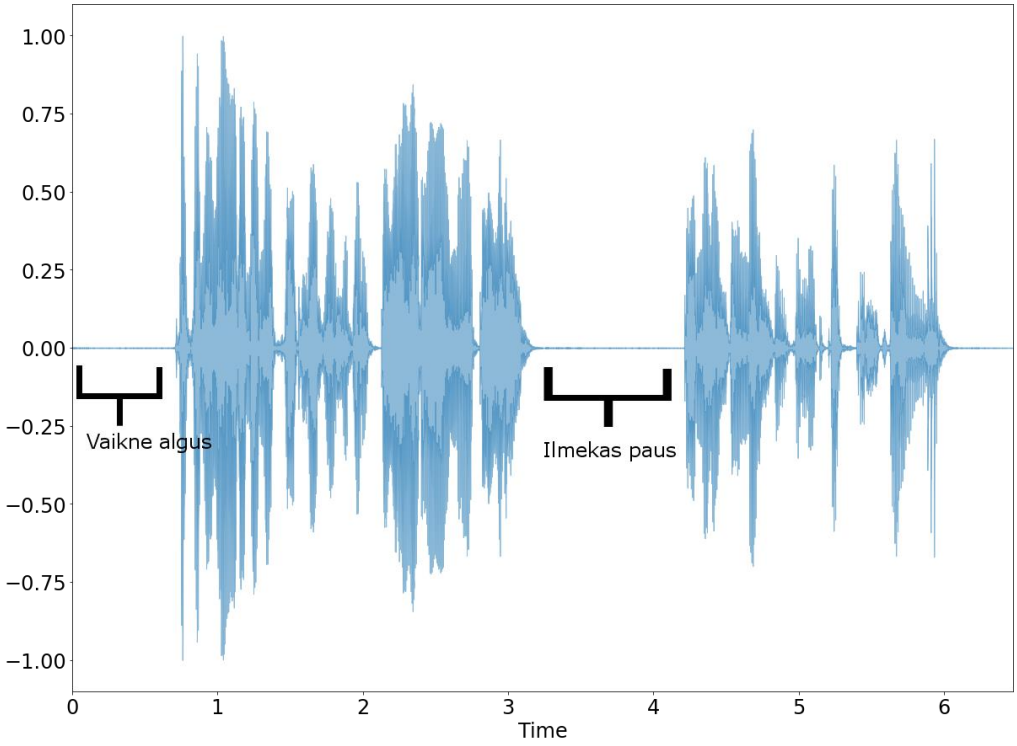
EKI ilukirjanduse kõnekorpus[11] [19] sisaldab meessoost diktori Meelis Kompuse häält ja naissoost näitleja Külli Reinumäe häält. Meelis loeb ette 9959 lauset Tammsaare „Tõest ja õigusest“, mis moodustab 18,5 tundi audiot. Külli loeb ette 6183 lauset erinevatest kaasaegsetest romaanidest, mis moodustab 8 tundi audiot. Seega kokku loetakse 16 142 lauset ehk 26,5 tundi audiot. Meelise ja Külli audio mahu erinevus võib mõjutada kõnesünteesi mudelite tulemusi, kuna rohkem näiteid läbivaadanud hääl võib olla parema kõlaga ning teha vähem vigu.

EKI ilukirjanduse kõnekorpuse helifailide transkriptsioonid on enamasti kirjutatud häälduspäraselt. Kuna tegu on ilukirjandusega, ei sisalda laused palju lühendeid, mille hääldus erineb kirjapildist, ning numbrid on juba sõnadena kirjutatud. Kuigi transkriptsioonid on häälduspärased, siis sisaldavad need palju erisümboleid, näiteks vanu jutumärke, mis ei mõjuta lausete hääldust. Et lihtsustada häälikute leidmise protsessi, võiks selliseid erisümboleid töötlemise ajal välja filtreerida.

EKI ilukirjanduse kõnekorpus on müravabas keskkonnas lindistatud, seega ei sisalda helifailid taustamüra ning hiireklikke. Siiski on tegu ilukirjandusega, seega leidub helifailides iseloomulikke omadusi. Näiteks võrreldes uudiste lugejatega, teevad Meelis ja Külli rohkem pikki ning ilmekaid pause. Lausete keskel asuvate pauside pikkus on varieeruv, seega paljud pausid ei kesta üle poole sekundi, kuid leidub ka üle sekundi kestvaid pause. Selliste pikkade pauside mõju Deep Voice 3 mudelitele võib olla kahjulik, kuna mudelitel on raske ennustada pauside pikkuseid, seega võiks eeltöötluse üks tulemusi olla pikkade pauside kärpimine.

Joonisel 7 on EKI ilukirjanduse kõnekorpusest pärit lause „Aga mina arvan, et need hobused on Pearu oma töö: tema ise lasigi nad koplist välja,“ helisignaali kujutis. Lause on lugenud Meelis ning joonisel on näha helifaili alguses müravaba pausi ning helifaili keskel umbes sekundilise kestvusega pausi.

[11] Allalaaditav aadressil https://www.eki.ee/litsents/

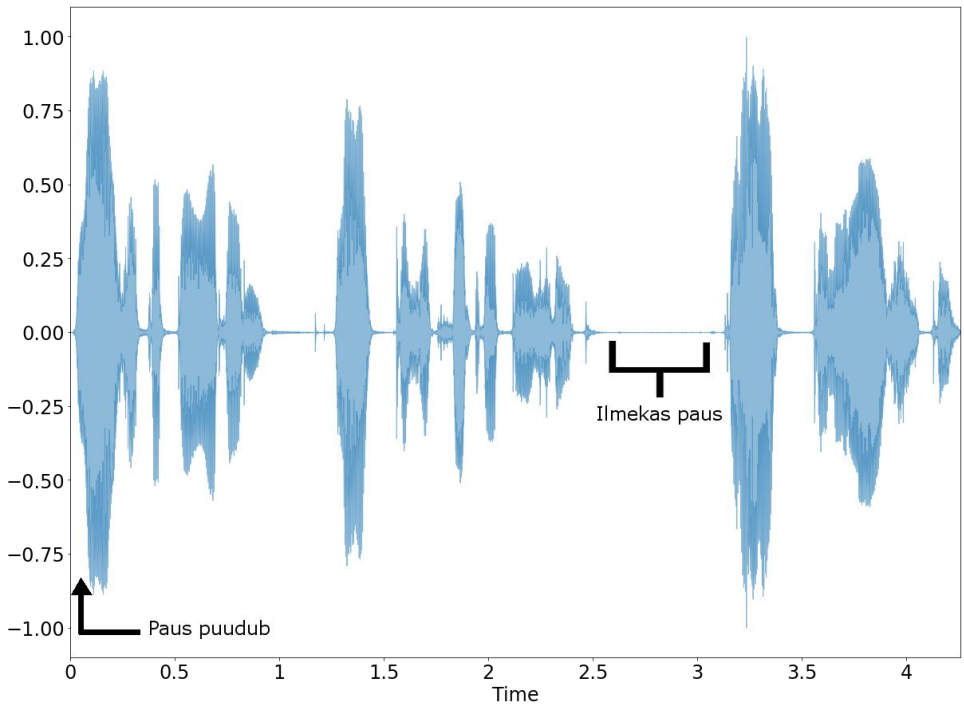


Joonis 7. EKI ilukirjanduse kõnekorpusest pärit lause „Aga mina arvan, et need hobused on Pearu oma töö: tema ise lasigi nad koplist välja,“ helisignaal. Lause lugeja on Meelis. X-teljel on aeg ja Y-teljel on signaali amplituud.

Peale ilmekate pauside iseloomustab ilukirjanduse lugemist ilmekas kõneviis. Seda on märgata eriti Meelise puhul, kes teeb naistegelaste otsekõnet kõrge häälega. Hääletooni muutusi otsekõne puhul on kuulda ka Külli puhul, kuid Külli hääletoon ei muutu nii suurel määral. Hääletooni muutused võivad mõjutada sagedustel põhinevaid helitöötluse meetodeid, kuna hääletoonist sõltub kõne sagedus. See võib mõjutada näiteks mel-spektrogrammi, mida kasutatakse Deep Voice 3 süsteemis mudelite treenimisel. Kuna Meelise häält sünteesiv mudel saab treeningandmeteks ilmekat kõne, siis mudeli täpsus võib eeltöötlusest sõltumata langeda, kuna treenimiseks antud mel-spektrogrammid erinevad üksteisest rohkem kui tavaliselt. Samas võib oletada, et hääletooni muutused ei mõjuta energia ja nullpunktimääral põhinevaid tunnuseid märgatavalt, sest toonist sõltumata on inimkõne energia piisavalt kõrge ja helitute häälikute puhul on ZCR kõrge.

Suurim erinevus Meelise ja Külli helifailide vahel on failide alguses olev paus. Meelise helifailid algavad üldjuhul umbes pool sekundit kestva vaikusega, mida on näha ka joonisel 7, samas Külli helifailid algavad alla 0,1 sekundit kestva lühikese pausiga, kui sedagi. Seega ei saa alati eeldada, et helifaili alguses on pikk vaikus, mille järgi saab vaikuse tunnuseid määrata.

Joonisel 8 on pauside illustreerimiseks kujutatud EKI ilukirjanduse kõnekorpusest pärit lause „Juustepiiril läikisid higipiisad, ta laup oli niiske,“ helisignaal. Lause on lugenud Külli ning joonisel on näha, et helifaili algul puudub märgatav paus. Siiski on signaal sarnane Meelise omale joonisel 7, kuna mõlemad helifailid on müravabad ja sisaldavad pikemaid pause.



Joonis 8. EKI ilukirjanduse kõnekorpusest pärit lause „Juustepiiril läikisid higipiisad, ta laup oli niiske,“ helisignaal. Lause lugeja on Külli. X-teljel on aeg ja Y-teljel on signaali amplituud.

Kokkuvõtvalt on EKI ilukirjanduse kõnekorpus vaikses keskkonnas lindistatud ning ei sisalda mehhaanilist müra. Samas tuleb kõnekorpuse töötlusel jälgida nii helifailide keskel olevaid ilmekaid pause kui ka helifaili alguses olevat pausi, mille pikkus sõltub lugejast.

2.3 Tartu Ülikooli uudiste kõnekorpus

UT uudiste kõnekorpus [6] on Tartu Ülikooli foneetika laboris lindistatud kõnekorpus, mis sisaldab kahe mees- ja kahe naistudengi häält. Meeshääled on andnud Albert ja Kalev ning naishääled on andnud Mari ja Vesta. Ette loetakse erinevaid eestikeelseid uudiseid, moodustades kokku 36 017 lauset ehk 65,9 tundi audiot. Mari, Alberti, Kalevi ja Vesta loetud audio pikkus on vastavalt 22,6 tundi, 21 tundi, 16,8 tundi ja 5,5 tundi. Eri ettelugejate audio mahu erinevus võib sellel kõnekorpusel samuti tulemusi mõjutada.

Kuna tegemist on uudistega, siis UT uudiste kõnekorpuse helifailide transkriptsioonid sisaldavad nii numbrilisel kujul numbreid, firmade ning riikide lühendeid kui ka võõrnimesid, mille hääldus ja kirjapilt võivad erineda. Lisaks sisaldavad mõned uudised veebilehtede aadresse, mille hääldamine võib sõltub lugejast. Et teada, milliseid häälikuid helifailis loetakse, võiks transkriptsioonides olevad numbrid, lühendid ja võõrnimed eeltöötluse ajal häälduspärasemalt kirjutada.

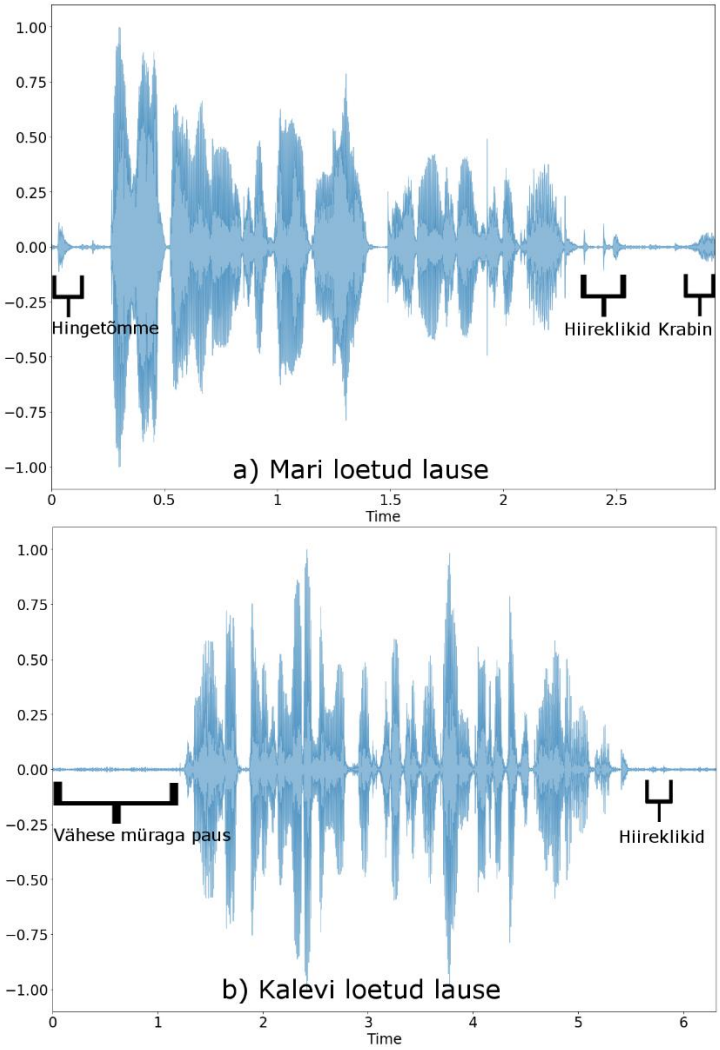
Võrreldes ilukirjanduse kõnekorpusega, puuduvad UT uudiste kõnekorpuses ilmekad pausid ja sage hääletooni muutmine. Selle asemel on kuulda helifailides rohkem pidevat müra, suu häälitsusi ja hiireklikke, mis on tekkinud salvestuse alustamisel ja lõpetamisel.

Pideva müra tugevus ei tundu sõltuvat lugejast, kuid hiireklikkide valjus ja asukoht helifailis muutub vastavalt lugejale. Mari loetud lausete juures on hiireklikk teiste lugejatega võrreldes valjem ning klikk tehakse kohe pärast lause lõppu. Seega ei ole Mari loetud helifailides pikka pausi kõne ja kliki vahel, mis võib raskendada eeltöötluse ajal kõne lõpu leidmist, kuna märgatavalt vali heli esineb helifailis kohe peale lause lõppu. Teised lugejad, eriti

Kalev, oletatavasti vajutavad hiirt pehmemalt ja jätavad aega lause lõpu ja kliki vahele, seega pole klikk nii vali ja äkiline.

Peale hiireklikkide erineb lugejate vahel samuti pausi pikkus enne lause lugemist. Võrreldes teistega, alustab Mari lause lugemist kiiresti peale helifaili lindistamise algust. Sellest tulenevalt võivad Mari helifailid alata kuuldava hingetõmbega, et kiirelt lause lugemist alustada. Teised lugejad jätavad tavaliselt vähemalt pool sekundit pausi enne lause lugemist. Seega peab eeltöötlusel arvestama, et kuigi tavaliselt on helifaili alguses paus, ei saa sellega alati arvestada.

Joonisel 9 on lugejate võrdlemise eesmärgil kujutatud Mari ja Kalevi loetud lauseid. Mari puhul on märgata helifaili alguses lühikest hingetõmbega täidetud ajahetke ning helifaili lõpus valjemaid hiireklikke koos mehhaanilise krabinaga. Kalev teeb enne lause lugemist pikema pausi ning tema hiireklikid pole nii märgatavad kui Mari omad. Siiski on mõlema lugeja puhul märgata, et helifailis on pauside ajal rohkem pidevat müra kui EKI ilukirjanduse kõnekorpuses.



Joonis 9. UT uudiste kõnekorpusest pärit lausete helisignaalid. X-teljel on aeg ja Y-teljel on signaali amplituud. a) Mari loetud lause „Tänapäeval on see tõug näituse ja seltsikoer," b) Kalevi loetud lause „Suures jalgpallis on värava mõõtmed 73 x 24 meetrit."

Kokkuvõtvalt on UT uudiste kõnekorpus lindistatud vähese pideva müraga keskkonnas, mille eeltöötlusel tuleb arvestada mehhaaniliste klikkide ja häälitsustega. Sarnaselt EKI kõnekorpusele, tuleb jälgida helifaili alguses olevat pausi, mille pikkus sõltub lugejast.

2.4 Eesti Rahvusringhäälingu raadiouudiste kõnekorpus

ERR raadiouudiste kõnekorpus [20] sisaldab seitsme ERR uudistelugeja häält. Uudistelugejateks on viis meest (Vallo, Tõnu, Tarmo, Meelis'[12] ja Indrek) ning kaks naist (Birgit ja Kai). Kokku on loetud 11 016 lauset ehk 14,8 tundi audiot. Ette loetakse erinevaid uudiseid. Kuna raadiouudiste korpust ei kasutata mudelite treenimiseks, pole täpne lausete jaotus lugejate vahel tähtis.

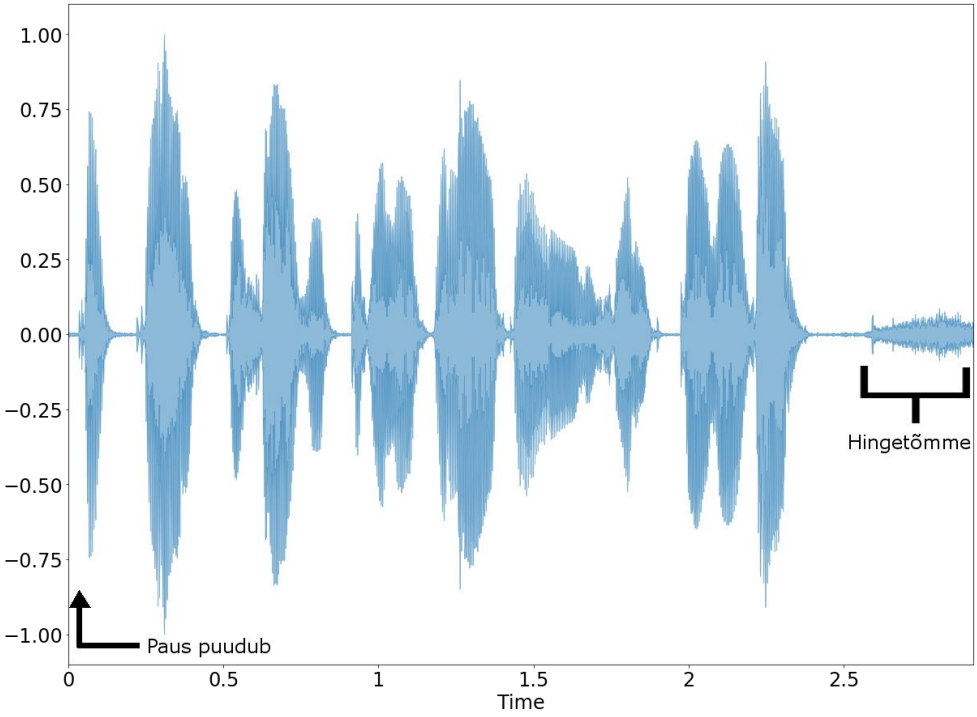
ERR raadiouudiste kõnekorpuse helifailide transkriptsioonid on kirjutatud väga häälduspäraselt. Erinevalt teistest kõnekorpustest, kajastuvad ERR raadiouudiste kõnekorpuste transkriptsioonides ka lugejate ekslikud sõnakordused ning kuigi tegu on uudistega, on numbrid ning kellaajad lahti kirjutatud. Siiski pole kõik häälduspäraselt kirjutatud, sest lühendite kirjapilt on jäetud lahti kirjutamata. Näiteks pole kirjas transkriptsioonis, kas HIV hääldus on tähthaaval lausutud „haa-ii-vee" või kiirelt lausutud „hiv".

ERR raadiouudiste kõnekorpuses pole erinevate lugejate vahel märgatavaid erinevusi kuulda. Eeldatavasti on iga lugeja helifailid saadud kogu raadiouudise tükeldamisel, mitte iga helifaili eraldi lindistamisel. Kuna raadiouudise ettelugemisel on tavaks kogu uudis kiiresti ning pause tegemata ette lugeda, siis saab iga lugeja korral eristada kolme liiki helifaile: uudiste algused, mis algavad lühikese pausiga ja lõppevad hingetõmbega, uudiste keskosad, mis algavad ilma pausita ja lõppevad hingetõmbega ning uudiste lõpud, mis algavad ilma pausita ja lõppevad lühikese pausiga.

Lühikesed või puuduvad pausid helifailide algul võivad raskendada vaikuse äratundmist eeltöötlusel, sest ei saa eeldada, et helifail algab vaikusega ning leida helifaili algusest vaikust kirjeldavaid tunnuseid. Eeltöötlust raskendavad ka valjud hingetõmbed, sest nende energia on piisavalt suur, et eristuda vaikusest, ning hingetõmmete ZCR väärtus on samuti kõrgem vaikuse omast.

Kõige rohkem esineb korpuses uudiste keskosi, seega joonisel 10 on keskosade illustreerimiseks kujutatud ERR raadiouudiste kõnekorpusest pärit lause „Kokkuleppe saavutamine pole aga endiselt kindel," helisignaal. Lause on lugenud Birgit ning joonisel on näha, et helifaili alguses puudub märgatav paus. Antud paus kestab veel vähem kui EKI ilukirjanduskorpuse lugeja Külli paus. Lisaks on näha helifaili lõpus hingetõmmet, mis on vähemalt sama vali kui UT uudiste korpuse hiireklikk, kuid palju pikema kestvusega.

[12] Kuna ERR raadiouudiste kõnekorpuse Meelis on sama Meelis, kes on EKI ilukirjanduse kõnekorpuses, siis erinevuse märkimiseks kasutatakse ERR raadiouudiste Meelise puhul eristamiseks '-märki.



Joonis 10. ERR raadiouudiste kõnekorpusest pärit lause „Kokkuleppe saavutamine pole aga endiselt kindel,“ helisignaal. Lause lugeja on Birgit. X-teljel on aeg ja Y-teljel on signaali amplituud.

Kuna suur osa ERR kõnekorpuse helifailidest algavad koheselt ja helifailide keskel on vaikus minimaalne, ei ole eeltöötlus vaikuse eemaldamiseks väga tähtis. Siiski saab kõnekorpusel eeltöötlust rakendades teada, kas eeltöötlus suudab eemaldada hingetõmbeid. Lisaks saab kontrollida, et eeltöötlus ei lõikaks kõne algust või lõppu liiga järsult raadiouudiste keskkonnas ära.

3. Eeltöötluse implementatsioon

Töö käigus on vaikuse eemaldamiseks loodud kaks erinevat eeltöötluse meetodit. Meetodeid kasutavat programmi saab vaadata valminud programmi repositooriumis[13]. Järgnevalt kirjeldatakse täpselt, mis on eeltöötluse eesmärk, mis töövahendit kasutatakse eeltöötluse jaoks ning mis on loodud eeltöötluse meetodite tööpõhimõte.

3.1 Eeltöötluse eesmärk

Et implementatsioonide detaile saaks paremini põhjendada, tuuakse välja antud eeltöötluse eesmärgid. Lisaks kirjeldatakse tingimusi, mida implementatsioon peab täitma.

Eeltöötluse põhieesmärk on eemaldada kõnekorpuste helifailide alguses ja lõpus olevaid vaikseid osasid. Vaiksete osade alla arvestatakse ka taustamüra, köhatusi ning hiireklõpse, seega kõike, mis pole helifailis etteloetud inimkõne. Samas ei tohi eeltöötluse käigus helifailidest kaduda kõne, seega ei tohi vaikuse eemaldamine olla liiga agressiivne. Kuna liiga agressiivselt eemaldatud kõne võib teha helifaili arusaamatuks, sest võidakse eemalda sõna algus, on parem eemaldada vaikust ettevaatlikult ja pigem jätta vaikus eemaldamata.

Peale otspunktide peab eeltöötlus eemaldama vaikust ka helifailide keskelt. Erinevalt otspunktidest võib helifailide keskel olev vaikus mõningal määral alles jääda, et säilitada helifailides olevat ilmekust. Vastasel juhul ei pruugi eeltöödeldud kõnekorpusel treenitud kõnesünteesi mudel kirjavahemärkide korral pause teha. Seega piisab, kui helifailide keskel olevatele vaikustele seada maksimaalne pikkus ja lõigata liiga pikad vaikused lühemateks. Samuti võiks keskmisi vaikusi eemaldada leebemalt kui otspunktide vaikusi, sest helifailide keskelt eemaldatud häälikud võivad mõjutada kõnesünteesi mudeleid rohkem kui varieeruva pikkusega vaikus.

Eeltöötluse implementatsioon, mis täitab eelnevaid eesmärke, peab samuti olema piisavalt automaatne, et implementatsiooni rakendamiseks ei pea kasutaja iga helifaili kohta sisestusi tegema. Kuna eeltöötluse rakendamine uue parandatud kõnekorpuse loomiseks on ühekordne tegevus, siis võiks kasutajalt siiski eeldada vähesel määral aega implementatsiooni rakendamiseks, näiteks peaks kasutaja sisestama kõnekorpuse kaustade asukoha. Samuti ei pruugi kõik kõnekorpused olla sama struktuuriga nagu eeltöötluse arendamisel kasutatud kõnekorpused, seega kasutaja peaks sellistel juhtudel koodi modifitseerima.

Lisaks automatiseeritusele, peab eeltöötluse implementatsiooni saama kergesti arvutusklastris rakendada. Suuri kõnekorpuseid on kiirem töödelda arvutusklastrites kui personaalarvutites, seega juba arvutusklastrites olevatele tööriistadele, nagu Python, toetuv implemenatsioon sobib hästi. Kuna arvutusklastrites olevad tööriistad on enamasti ka personaalarvutites, saab implementatsiooni kasutada ka personaalarvutites.

3.2 Librosa

Et täita eeltöötluse eesmärke, on valitud eeltöötluse põhiliseks tööriistaks Pythoni pakett librosa [21]. Tegemist on audiole keskenduva paketiga, mis suudab muuhulgas leida helist tunnuseid, transformeerida heli ja graafiliselt kujutada heli.

Kuna librosa on Pythoni pakett, saab seda rakendada Pythoni failides või Jupyter Notebookis, milles olevaid muutujaid on kasutajal kerge muuta. Seega on võimalik täita automatsiooni tingimust. Lisaks on Python kasutusel paljudes arvutusklastrites, seega on täidetud ka klastrites rakendamise tingimus. Töös on kood kirjutatud Jupyter Notebookis, sest nii on meetodeid lihtsam arendada ja graafiliselt kujutada.

[13] https://github.com/AndreasTeder/EST_lang_corpora_preprocessing

Librosal on paar eelist võrreldes teiste matemaatiliste Pythoni pakettidega, nagu SciPy[14]. Kuna librosa keskendub audiole, siis on mitmeid käske, millega saab kiiresti tunnuseid, nagu RMSE[15] ja ZCR[16], arvutada. Neid tunnuseid saaks arvutada ka SciPy abil, kuid sama tulemuse saavutamiseks peaks kirjutama rohkem koodi. Peale tunnuste leidmise on librosas tehtud lihtsaks heli graafiline kujutamine[17], mis on kasulik meetodite arendamisel ja testimisel.

Varasemalt mainitud helitöötlusprogramm Audacity on samuti alternatiiv librosale. Kasutades Audacity pistikprogrammi[18], võib Audacity meetodeid kutsuda välja läbi Pythoni skripti, täites automatsiooni tingimuse. Siiski vajab selline lahendus Audacity installeerimist arvutusklastrisse ning Audacity hoiatab, et sellise lahenduse puhul ei teatata alati kasutajale, miks midagi ei töötanud. Kuna mugavam oleks teha eeltöötlust ilma programme installeerimata ja teada, mis programmis toimub, eelistatakse töö käigus librosat.

3.3 RMSE ja ZCR baasil eeltöötlus

Esimene loodud meetoditest kasutab energiapõhist RMSE tunnust vaikuse märgendamiseks ning parandab leitud märgendeid ZCR abil. Meetodi idee aluseks on võetud L. R. Rabineri ja M. R. Samburi [1] kõne otspunktide leidmise algoritm, millele on lisatud transkriptsiooniga arvestamise mehhanism, helifaili alguses oleva vaikuse kontroll ja helifaili keskel oleva vaikuse eemaldamine. Järgnevalt kirjeldatakse, kuidas loodud meetodi tunnuseid leitakse ja kasutatakse ning kuidas leiti kasutatud parameetrid.

3.3.1 Ettevalmistus tunnuste leidmiseks

Meetodi rakendamiseks peab teadma, kus asuvad helifailid, mida on vaja töödelda. Töös kasutatud korpused sisaldavad iga lugeja kohta .csv laiendiga faili, kus on kirjas nii helifailide asukohad kui ka vastavad transkriptsioonid. Meetod salvestab helifailide asukohad ja transkriptsioonid ning alustab nende töötlemist.

Transkriptsioonide häälduspärasemaks muutmiseks on kasutusel I. Heina ja L. Rätsepa loodud teksti eeltöötleja[19], mida rakendatakse tööga seotud kõnesünteesi projektis. Teksti eeltöötleja kirjutab muuhulgas tekstis esinevad lühendid, numbrid, kuupäevad ja veebilehtede aadressid häälduspäraselt sõnadena välja. Näiteks muudetakse lause „Aastal 2020 oli minu palk 110 eurot“ lauseks „Aastal kaks tuhat kakskümmend oli minu palk sada kümme eurot“. Teksti eeltöötluse rakendamise järel eemaldatakse parandatud tekstidest kõik sümbolid, mis pole tähed ega numbrid, sest nad ei mõjuta lause häädlust ja nii on võimalik lihtsalt leida, kas lause lõppeb helitu või helilise häälikuga.

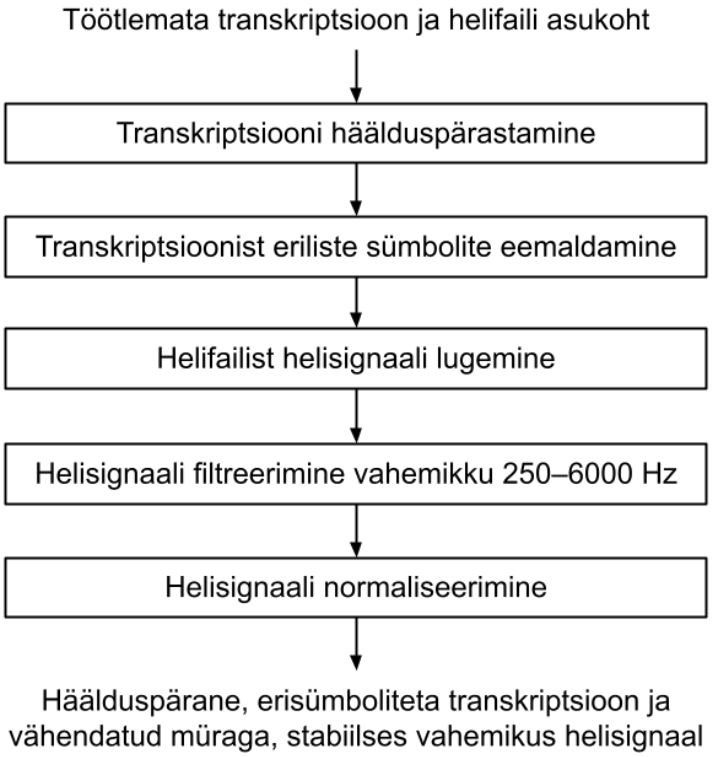
Pärast transkriptsioonide parandamist, töötleb meetod transkriptsiooniga seotud helifaili. Esmalt loetakse sisse töödeldav helifail diskreetimissagedusega 22 050 Hz ja saadud helisignaalile rakendatakse nii kõrg- kui ka madalpääsfiltrit. Filtrite rakendamise tulemusena vähendatakse helisignaalis alla 250 Hz ja üle 6000 Hz sagedusega sageduskomponentide magnituude drastiliselt. Selliste filtrite rakendamisel jätame valdavalt alles inimhääle, kuid eemaldame väga madala ja kõrge sageduse müra. Sageduste vahemik on valitud J. Sheni jt [22] näitel. Samas on loodud algoritme, mis kasutavad 100 kuni 4000 Hz vahemikke [1],

[14] https://scipy.org/
[15] https://librosa.org/doc/main/generated/librosa.feature.rms.html
[16] https://librosa.org/doc/0.8.0/generated/librosa.feature.zero_crossing_rate.html
[17] https://librosa.org/doc/main/generated/librosa.display.specshow.html
[18] https://manual.audacityteam.org/man/scripting.html
[19] https://github.com/TartuNLP/tts_preprocess_et

kuid katsetamise käigus tundus, et inimkõne võib olla üle 4000 Hz ning on harva alla 250 Hz sagedusega.

Viimaks rakendatakse filtreeritud helisignaalile normaliseerimist[20], mille tulemusena helisignaali diskreetide amplituudid viiakse vahemikku -1 kuni 1. Ilma normaliseerimata on diskreetide amplituudid suuremas sõltuvuses lugejast ja mikrofonist, sest lugejad ja mikrofonid pole alati sama valjud, seega salvestatud diskreetide amplituudid on erinevates vahemikes. Normaliseerimine aitab vähendada heli salvestuskeskkondade erinevusi, seega saab meetodis kindlamalt kasutada helifailist sõltumatuid konstante.

Joonisel 11 on kujutatud protsessi sammud, kuidas algsest transkriptsioonist ja helifaili asukohast saadakse ettevalmistatud transkriptsioonid ja helisignaal.



Joonis 11. RMSE ja ZCR baasil eeltöötluse ettevalmistuse etapid.

Ettevalmistuse tulemusena on meetod saanud filtreeritud ja normaliseeritud helifaili signaali, mille põhjal saab tunnuseid arvutada. Signaalil on juures ka eeltöödeldud transkriptsioon, mille põhjal saab viimaseid häälikuid leida.

3.3.2 RMSE tunnuse leidmine ja kasutamine

Kui meetodil on ettevalmistused tehtud, saab alustada tunnuste leidmise ja vaikuse märgendamisega. Et leida RMSE väärtusi, raamistatakse helisignaal ja iga raami kohta arvutatakse RMSE väärtus. Raamidele lisatakse kattuvust, nõnda võib raami mahutada piisavalt palju diskreete, et kõrge amplituudiga erandid ei mõjutaks tulemust, ning samas võib raamide vahe olla piisavalt väike, et täpselt määrata vaikuse asukohta. Raami äärtes olevate diskreetide osakaalu vähendamiseks rakendatakse akna funktsiooni.

[20] https://librosa.org/doc/main/generated/librosa.util.normalize.html

Kuigi librosas puudub otsene meetod signaali raamistamiseks ja aknastamiseks, rakendatakse nii raamistamist kui ka aknastamist lühiaja Fourier' pöörde leidmisel[21]. Lühiaja Fourier' pöörde leidmiseks on vaja teada raami suurust diskreetides (antud juhul 512) ja raamidevahelist kaugust diskreetides (antud juhul 128). Librosa rakendab lühiaja Fourier' pöörde leidmisel vaikimisi Hanni akent, seega lühiaja Fourier' pöördest saadud magnituudid on juba töödeldud. Viimaks rakendatakse RMSE leidmise valemit, mis ei arvuta RMSE väärtust raami diskreetide amplituudide järgi, nagu standardselt, vaid raami sageduskomponentide magnituudi järgi. Muutusele vaatamata hindab RMSE valem sama väärtust, kuna sageduskomponentide magnituudid mõõdavad raamis olevat energiat sageduse perspektiivist.

Teades RMSE väärtust iga raami kohta, saab meetod alustada vaikuse märgendamist. Et vaikust ja kõne eristada, kasutatakse märgendamisel läve, mida ideaalis võiks arvutada dünaamiliselt helifaili alguses oleva vaikse vahemiku pealt. L. R. Rabiner ja M. R. Sambur [1] eeldasid, et helifailide alguses on 100 ms vaikust, kuid näiteks Külli helifailide puhul see eeldus ei kehti. Seega kontrollib meetod esiteks, kas helifail algab vaikusega, ning vaikuse puudumisel kasutatakse edaspidi meetodi lihtsustatud versiooni, mille lävi ei sõltu vaikusest.

Kuna programmi luues optimeeriti enne lihtsustatud versiooni parameetrid ja seejärel loodi dünaamilise lävega uuendatud versioon, puudub lihtsustatud versioonis kaks sammu. Esiteks ei rakendata lihtsustatud versioonis madal- ja kõrgpääsfiltreid, seega peab ka RMSE väärtused uuesti arvutama, sest helifaili energia on muutunud. Teiseks ei kasutata lihtsamas versioonis ZCR väärtusi tulemuse täpsustamiseks. ZCR väärtustega arvestamist prooviti lisada, kuid erinevate kõnelejate ZCR väärtused ei olnud piisavalt stabiilsed, et neid saaks ilma näidisvaikuseta, mille pealt saaks leida ZCR keskmist väärtust, kasutada.

Dünaamiline meetod kasutab läve leidmiseks helifaili vahemikku 0,1 sekundist 0,2 sekundini. Kuna helifail võib alata kliki või muu mehhaanilise müraga, näiteks kõneleja Mari puhul, alustatakse läve leidmist alates 0,1 sekundist. Kuigi müra võib kesta rohkem kui 0,1 sekundit, peab arvestama peatselt algava kõnega, mistõttu ei saa vahemikku liiga kaugele liigutada. Sellest vahemikust saadud RMSE väärtuste põhjal otsustatakse esmalt, kas kasutada lihtsustatud meetodit. Kui vahemiku RMSE väärtuste aritmeetiline keskmine on kõrgem kui 10% kogu helifaili maksimaalsest RMSE väärtusest, oletatakse, et helifail algab kõnega, ja kasutatakse lihtsustatud versiooni.

Läve leidmiseks kasutatakse L. R. Rabineri ja M. R. Samburi [1] läve valemit, mida on katsetamise käigus ülesandele sobivamaks muudetud. Läveks võetakse 6% kogu helifaili maksimaalsest RMSE väärtusest, millest on lahutatud vaikuse vahemiku RMSE väärtuste aritmeetiline keskmine. Lihtsustatud koodi kujul näeb läve arvutamine välja järgnevalt:

```
RMSE_lävi = 0.06 * (max(RMSE) - average(RMSE_vaikne_vahemik))
```

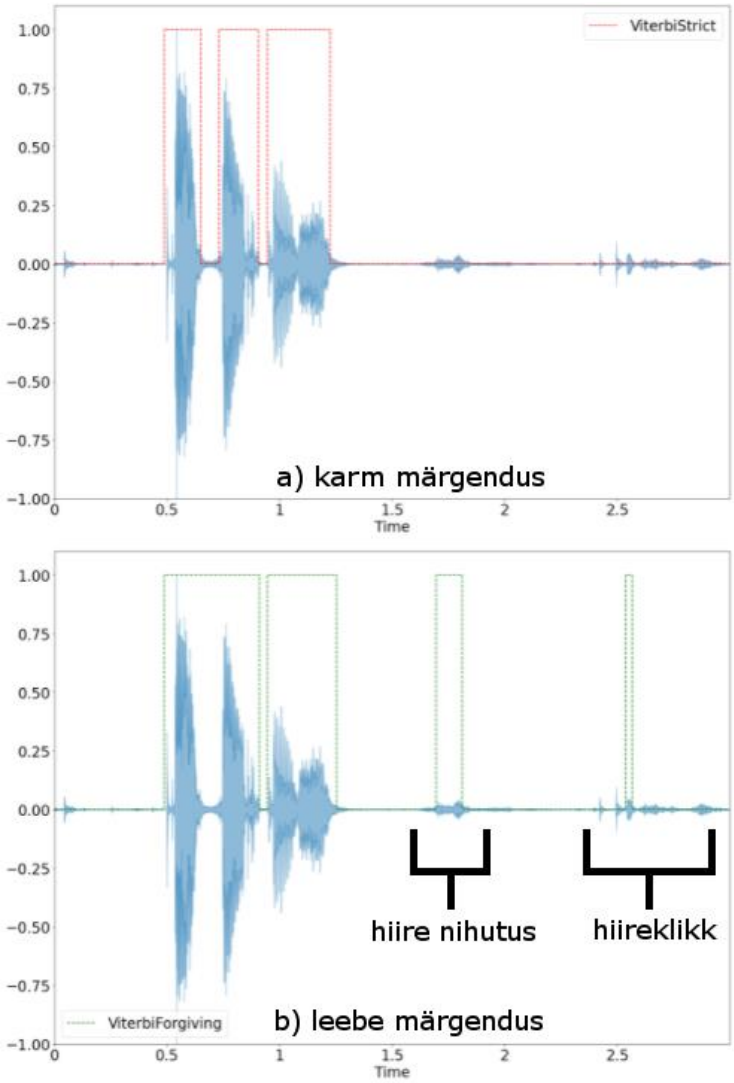
Teades RMSE väärtuste läve, millest madalamat RMSE väärtust saab märgendada vaikuseks, võib raame märgendada ainult RMSE väärtuse ja läve võrdlusel ning saada korrektseid tulemusi. See meetod ei ole siiski ideaalne, sest nii lihtne lähenemine ei arvesta üksikute raamidega, mille RMSE väärtus on äkitselt väiksem kui lävi. Näiteks võib kõneleja hääl sõna lõpus langeda või kõneleja hakkab poole sõna peal kokutama. Lihtne lähenemine märgendaks neid juhte vaikusena, kuigi tegelikult on veel tegu kõnega. Kuna L. R. Rabiner ja M. R. Sambur [1] määrasid ainult kõne otspunkte, on nende lahendust raske tööülesandele vastavaks muuta.

[21] https://librosa.org/doc/0.8.0/generated/librosa.stft.html

Loodud meetodis kasutatakse vaikuse märgendamiseks Markovi peitmudelit, mille seisunditeks on vaikus ja kõne. Eestikeelsete terminite leidmiseks Markovi peitmudeli kohta on kasutatud H. Läänemetsa magistritööd [23]. Kui mudeli seisundite jada pikkuseks on helifaili raamide arv, siis Viterbi algoritmiga leitud parim seisundite jada on samuti vaikuseks ja kõneks märgendatud raamide jada. Kuna helifailis järgneb kõnele üldjuhul kõne, suurendatakse kõne seisundisse jäämise tõenäosust, see lahendab eelmises lõigus tõstatatud probleemi, kus kokutamised ja kõne lõpp võidakse vaikusena märgendada. Kõne seisundi vaatluste tõenäosuseks kasutatakse RMSE väärtusi, mis on muudetud tõenäosusteks, lahutades raamide RMSE väärtustest läve väärtuse ja kujutades saadud tulemusi 0 kuni 1 vahemikus. Seega kõrge RMSE väärtusega raamil on suurem kõne tõenäosus ning kui RMSE väärtus on lävega võrdne on kõne tõenäosus 50%. Olles määranud vajalikud Markovi peitmudeli komponendid, saab rakendada Viterbi algoritmi ja iga raami kohta määrata, kas tegu on vaikuse või kõnega. Markovi peitmudelite loomiseks koodis on kasutatud librosa „*Viterby decoding*“ näidet [24].

Vaikuse märgendamisel annaks parima tulemuse, kui hiireklikid ja mehhaaniline müra helifaili otspunktides loetakse vaikuseks, samas vaiksed sõnade lõpud ja algused kõne keskel loetakse kõneks. Kuna ühe lävega on mõlema tingimuse täitmine ebatäpne, kasutatakse kahekordset märgendamist. Otspunktide vaikuse märgendamiseks kasutatakse karmimat märgendamist ja lausete keskel olevate pauside märgendamiseks leebemat märgendamist. Karmima märgendaja korral on RMSE läve väärtust suurendatud ja Markovi peitmudelis on kõne seisundisse jäämise tõenäosust vähendatud, seega lühiajalist müra märgendatakse kõneks harvemini.

Joonisel 12 on kujutatud loodud märgendajate võrdlus, kus on märgendatud UT kõnekorpusest pärit lause „Ähvardus!“ Kõnega raamid on väärtusega 1 ja vaikusega raamid on väärtusega 0. Joonisel on näha, et karm märgendaja ei tuvastanud hiire nihutust ja hiireklikki kõnena, kuid leebe märgendaja on müra valjemad osad kõneks märkinud. Samas ei tükelda leebe märgendaja helisignaali kõne osa nii palju kui karm märgendaja.

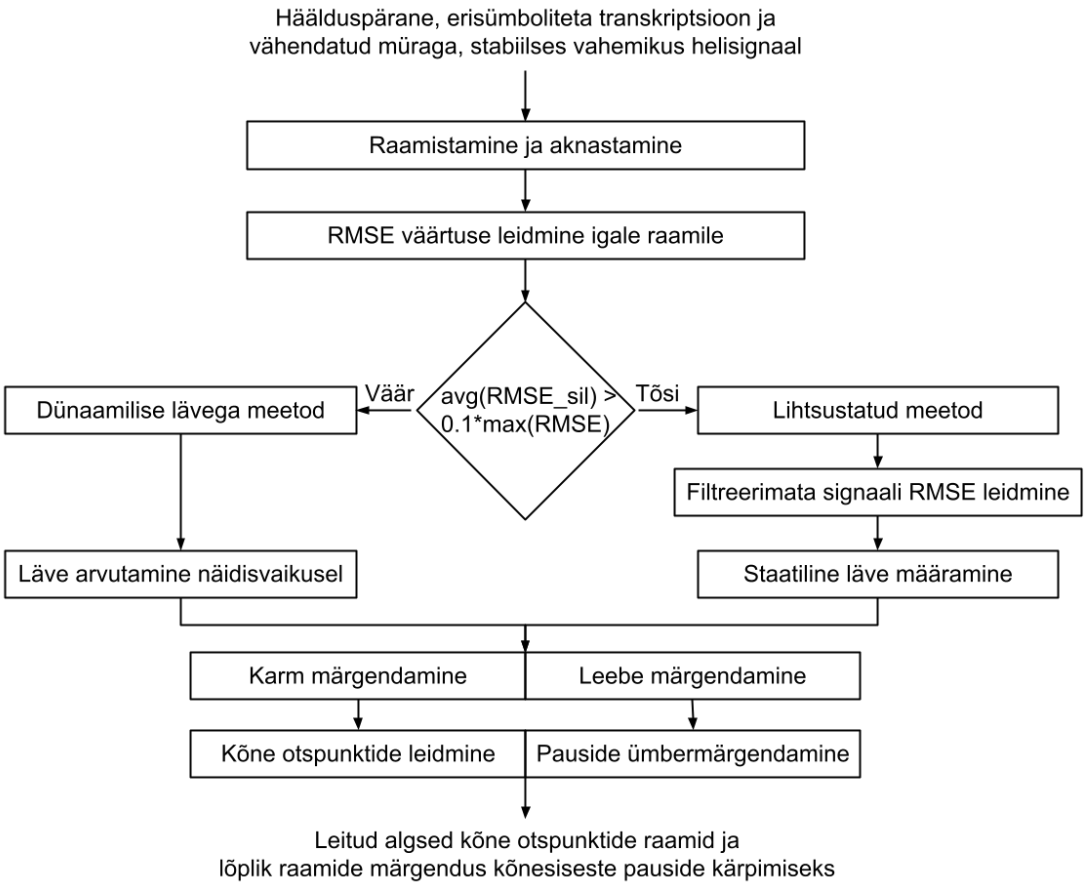


Joonis 12. Lause „Ähvardus!“ vaikuse ja kõne märgendus. X-teljel on aeg ja Y-teljel on signaali amplituud ning märgend (kõne väärtuseks on 1), kus märgendajatena on kasutatud a) karmi otspunktide märgendajat b) leebet lausesisest märgendajat.

Enne ZCR väärtuste leidmist on vaja läbida kaks viimast etappi: mõlema märgenduse töötlemine. Karmi märgendaja tulemustest leitakse kõne otspunktid, ehk millises raamis algab ning lõppeb kõne. Otspunktide leidmiseks vaadatakse kronoloogilises järjestuses läbi kõigi raamide märgendused ja salvestatakse esimene ning viimane kõne sisaldav raam. Kuna kohe helifaili alguses võib esineda lühikest müra, näiteks hingetõmme või hiireklikk Mari puhul, ei arvestata esimesi kõne märgendeid. Samuti kasutatakse A. Ganapathiraju jt [16] ideed defineerida kõne miinimumkestvus (antud juhul 0,07 sekundit). Kui kõne vahemik ei kesta piisavalt kaua, ei muudeta otspunkte, sest tõenäoliselt on tegemist ajutise müraga.

Leebe märgendaja tulemuste põhjal kärbitakse hiljem lausetesiseseid pause määratud maksimaalse pikkuseni (antud juhul 0,4 sekundit). Kärpimise jaoks vaadatakse samuti kronoloogilises järjestuses läbi raamide märgendused ning maksimaalsest pikkusest lühemad vaikused märgendatakse ümber kõneks. Kui vaikus on pikem maksimaalsest pikkusest, märgendatakse kõneks ainult piisav hulk raame vaikse tükki äärtest.

Joonisel 13 on kujutatud protsessi sammud, kuidas ettevalmistatud transkriptsioonist ja helisignaalist saadakse RMSE väärtused ning kuidas RMSE väärtusi kasutatakse vaikuse ja kõne märgendamiseks.



Joonis 13. RMSE ja ZCR baasil eeltöötluse RMSE tunnuse leidmise ja kasutamise etapid.

Pärast RMSE väärtuste rakendamist on leitud kõne otspunktide raamid. Lisaks on nende otspunktide vahele jäävad raamid juba märgendatud. Et kõne otspunktide täpsust parandada, saab kasutada ZCR tunnust.

3.3.3 ZCR tunnuse leidmine ja kasutamine

Erinevalt RMSE väärtuste leidmisest, ei saa librosas ZCR tulemusi helisignaali sagedus-komponentide abil leida, seega kasutatakse ZCR leidmiseks varasemalt eeltöödeldud helisignaali. Raamide ZCR väärtuste leidmisel kasutatakse sama raami suurust ja raamidevahelist kaugust, mida rakendati RMSE puhul, seega on raamid üksteisega vastavuses.

Sarnaselt RMSE tunnusele, kasutab meetod ZCR puhul helitute häälikute tuvastamiseks läve. Samuti kasutatakse läve leidmiseks sama helisignaali vahemikku, sest varasemalt on oletatud, et seal on vaikus. Läve leidmiseks kasutatakse L. R. Rabineri ja M. R. Samburi [1] läve valemit, mille kordajat on arendamise käigus muudetud, seega võetakse läveks vaikuse vahemiku ZCR väärtuse aritmeetiline keskmine, millele on liidetud sama vahemiku 2,1-kordne standardhälve. Lihtsustatud koodi kujul näeb läve arvutamine välja järgnevalt:

```
ZCR_lävi = average(ZCR_vaikne_vahemik) + 2.1 * std(ZCR_vaikne_va-
hemik)
```

L. R. Rabiner ja M. R. Sambur [1] kasutasid leitud läve, et laiendada kõne otspunkte helitu kõne korral. Selle jaoks vaadati otspunktide ääres 0,25 sekundit kestvas ajavahemikus, kas leidub kokku piisavalt palju raame, mille ZCR väärtus on lävest suurem. Kui sellised raamid leidusid, muudeti otspunktide väärtused viimasteks läve ületanud raamideks. Loodud

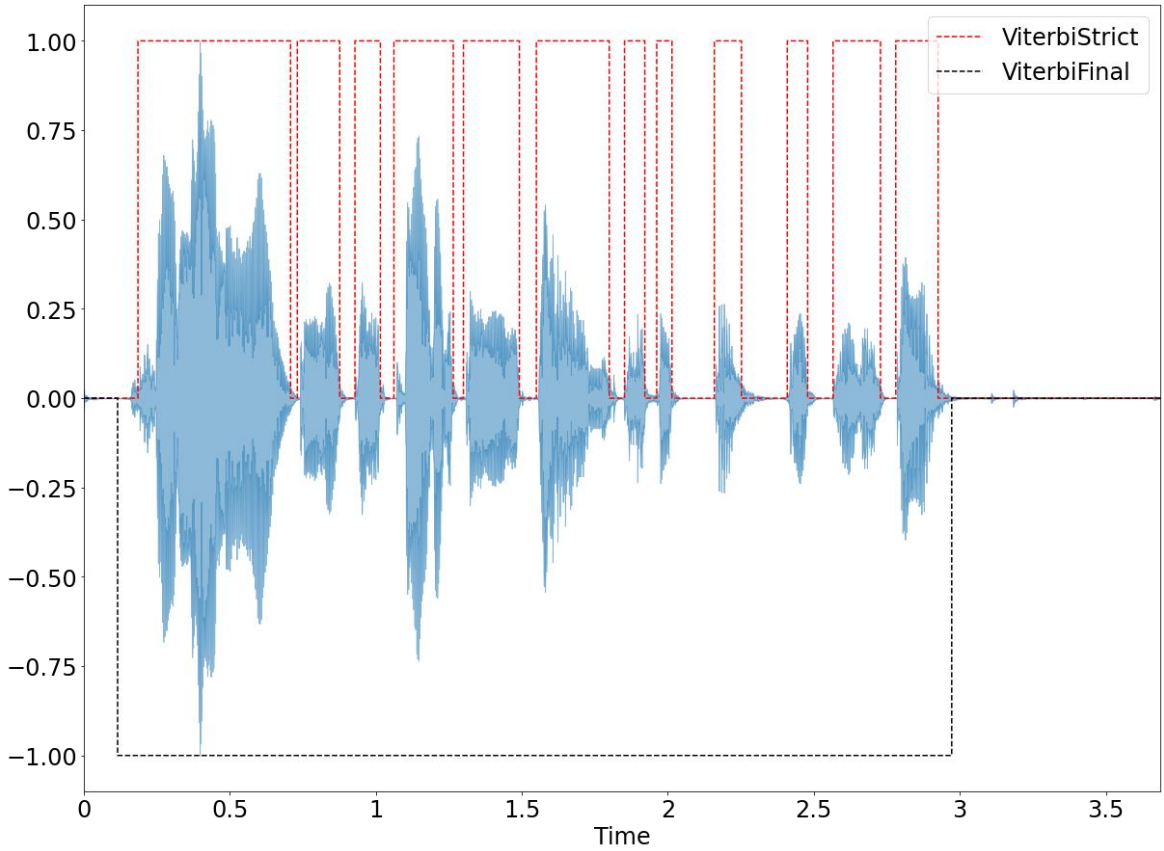
meetodis on eelnevat lähenemist muudetud, sest meetod saab kasutada transkriptsioone ning meetodis tahetakse vaikust ettevaatlikumalt eemaldada.

Esimese muudatusena ei arvestata ZCR väärtust kõne alguspunkti täpsustamisel. Kuna hingetõmmetel ja suumatsudel on kõrge ZCR väärtus, esines juhte, kus sellised defektid jäid töödeldud helifailidesse. Samas kõne alguspunkti täpsustamata võis töödeldud helifaili algus olla liiga järsk. Probleemi lahenduseks laiendatakse alguspunkti 0,07 sekundi võrra, sõltumata esimesest häälikust või ZCR väärtustest. Selline lähenemine annab piisavalt hea tulemuse, sest müra esineb töödeldud helifailide alguses harva ning lausete algused on üldjuhul sujuvad.

Teise muudatusena tehakse vaadeldava ajavahemiku pikkus 0,25 sekundi asemel transkriptsioonist sõltuvaks. Täpsemalt on ajavahemiku pikkuseks vaikimisi 0,12 sekundit ning transkriptsiooni viimase kahe hääliku põhjal võib vahemik suureneda. Lähenemise arendamise käigus tundus, et sulghäälikute ning „f“ ja „h“ tuvastamine transkriptsiooni viimase kahe hääliku seast on keeruline, seega sellistel juhtudel suurendatakse ajavahemikku 0,04 sekundi võrra. Seda reeglit ei rakendata juhtudel, kus viimane häälik on täishäälik, sest kõne lõppeb piisavalt valjult. Kui viimased kaks häälikut on „kk“, „pp“, „tt“, või „ll“, näiteks sõnade „kapp“ ja „kell“ juures, suurendatakse vahemikku samuti 0,04 sekundi võrra, sest neid häälikupaare oli raske tuvastada.

Viimase muudatusena laiendatakse kõne lõpppunkti peale ZCR väärtusega arvestamist veel lisaks. Lõpppunkti liigutatakse sõltumata viimasest häälikust 0,05 sekundi võrra, et kompenseerida karmi märgendaja agressiivsust. Et veel rohkem vähendada järsu lauselõpu tõenäosust, uuendatakse lõpppunkti asukohta iga läve ületanud ZCR väärtusega raami juures. Kuigi muudatuste tulemusel võib lähenemine jätta sisse harvadel juhtudel klikke ja muud müra, sisaldavad töödeldud helifailid vähem äkilisi lõppe, mis võiks kõnesünteesimisel olla eelistatud.

Kui uuendatud lähenemisega on kõne otspunktid täpsustatud, saab luua lõpliku vaikuse märgenduse, mille põhjal saab vaikseid raame eemaldada. Selleks märgendatakse vaikuseks kõik leebe märgenduse raamid kuni kõne alguspunktini ja alates kõne lõpppunktist. Et illustreerida otspunktide täpsustamist, on joonisel 14 kujutatud range ja lõpliku märgendamise võrdluseks lause „Samal ajal tuleks kordajaid vähemaks võtta", ütles ta,“ märgendamine. Lõpliku märgendamise juures on märgatav, et kogu kõne on ühes osas. See on soovitud tulemus, sest kõne sees pole ükski paus üle maksimaalse kestvuse. Samuti on märgatav, et lõpliku märgenduse otspunktid asetsevad laiemalt, mistõttu ei lõika need kõnne sisse, nagu juhtub range märgenduse puhul.



Joonis 14. Lause „Samal ajal tuleks kordajaid vähemaks võtta", ütles ta,“ range märgendus (punaselt) ja lõplik märgendus (mustalt). X-teljel on aeg ja Y-teljel on signaali amplituud ja märgend (kõne väärtuseks on 1 ja -1).

Viimase sammuna kasutatakse lõplikku märgendust ning luuakse uus helisignaal, kuhu salvestatakse ainult need diskreedid, mis on kõneks märgendatud raamide sees. Et vältida diskreetide kordumist, sest raamid on kattuvad, lisatakse kõnet sisaldavast raamist uude signaali ainult raamidevaheline kaugus diskreete. Lisatavad diskreedid on võetud raamide algusest (meetodi valmimise järel jõuti arusaamisele, et tegelikult annaks täpsema tulemuse raamide keskelt diskreetide lisamine). Pärast helisignaali loomist kirjutatakse see faili.

Joonisel 15 on kujutatud protsessi sammud, kuidas ZCR väärtused leitakse ning kuidas neid kasutatakse dünaamilise lävega versioonis. Kuigi lihtsustatud versioonis ei kasutata ZCR väärtust, siis transkriptsiooni viimaste häälikute põhjal laiendatakse kõne lõpppunkti. Seega ka lihtsustatud versioon arvestab helitute häälikutega, kuid tehtud vigade arv on suurem, sest lõpppunkti ei määrata nii täpselt.

Joonis 15. RMSE ja ZCR baasil eeltöötluse ZCR tunnuse leidmise ja kasutamise etapid dünaamilise lävega meetodi versioonis.

ZCR väärtustega arvestades on meetod täpsem helitute häälikute korral, eriti kuna arvestatakse helifaili transkriptsiooniga. Kuna tööülesande kohaselt eelistatakse liigset vaikust äkilistele helifaili algustele ja lõppudele, laiendab meetod algseid kõne otspunkte vastavalt meetodi arendamise käigus leitud parameetritele.

3.3.4 Parameetrite leidmine

Meetodis kasutatud parameetreid saab liigitada kahte gruppi. Esiteks on kasutusel teaduslikud parameetrid, mis on võetud varasematest uurimustest ja librosa dokumentatsioonist. Neid parameetreid pole meetodi arendamise käigus muudetud. Teiseks on kasutusel optimeeritud parameetrid, mille algseteks väärtusteks on kasutatud varasemaid uurimusi või juhuslikke väärtuseid ning arendamise käigus on väärtuseid täpsustatud.

Teaduslikud parameetrid selle meetodi juures on diskreetimissagedus, raami suurus ja raamidevaheline kaugus. Diskreetimissageduseks on valitud librosa vaikeväärtus[22] (22 050 Hz), sest G. Anbarjafari [25] järgi on selle Nyquisti sagedus 11 025 Hz, mis on kõrgeim sagedus helisignaalis, mis ei põhjusta diskreetmoonutust (ingl. *aliasing*). Kuna inimkõne suurim sagedus on väiksem kui 11 025 Hz [22], sobib librosa vaikeväärtus.

[22] https://librosa.org/doc/main/generated/librosa.load.html

Raami suuruseks ja raamidevaheliseks kauguseks on valitud vastavalt 512 ja 128 diskreeti, sest librosa järgi[21] sobivad need raamide parameetrite väärtused kõnetöötluseks. Kuna diskreetmissagedus on 22 050 Hz, siis iga raami kestvus on 23 ms ja raamidevaheline kaugus on 6 ms. Leitud raami kestvus sobib ka varasemate uuringute põhjal, sest A. Ganapathiraju jt [16] kohaselt sobib kõne raamistamiseks 20 ms kestev raam, mis on kasutatud raamiga samas suurusjärgus.

Võrreldes teaduslike parameetritega on meetodis optimeeritud parameetreid rohkem, näiteks lävede valemite kordajad, minimaalne kõne kestvus ja helitute häälikute hulk. Parameetrite algväärtused lävede juures on võetud vastavalt lävele [1], kuid ülejäänud algväärtused on juhuslikult võetud. Selliste parameetrite optimeerimiseks muudetakse ühte parameetrit ning töödeldakse iga kõneleja kohta 5 helifaili, mis jäid optimeerimiste käigus muutumatuks. Kui töödeldud helifailid tundusid varasemast parema kõlaga, jäeti parameetri muudatus sisse. Pärast paari edukat optimeerimist kontrolliti tulemust, kasutades iga kõneleja kohta 5 suvalist helifaili, et vältida ülesobitamist. Sellise protsessi käigus märgati lisaks erinevaid vigu, mida meetod saaks parandada, näiteks millised häälikud lõigatakse lõpust tihti ära ja mis vajavad seetõttu suuremat vahemikku ZCR väärtuse vaatamise juures. Alternatiivina oleks võinud luua helifailidest testhulga, kus iga helifaili vaiksed vahemikud märgendatakse käsitsi ära. Siis saaks parameetrite optimeerimisel kiiremini võrrelda testhulga helifaile töödeldud helifailidega, kuid helifailide käsitsi märgendamine oleks olnud liiga ajakulukas.

3.4 Autosegmenteerija baasil eeltöötlus

Teine loodud meetoditest kasutab akustiliste mudelite põhist autosegmenteerijat[23], mille on loonud T. Alumäe jt [26]. Järgnevalt kirjeldatakse autosegmenteerija tööpõhimõtet ning kuidas autosegmenteerijat saab rakendada vaikuse eemaldamisel.

3.4.1 Autosegmenteerija tööpõhimõte

Autosegmenteerija on veebirakendus, millega saab eestikeelset helifaili joondada vastava transkriptsiooniga. Andes rakendusele ette helifaili ja helifaili transkriptsiooni, tagastab rakendus Praati *TextGrid*[24] tüüpi faili, kus on kirjas nii sõnade, üksikute häälikute kui ka vaikuste algus- ja lõpphetk helifailis. Veebirakendus kasutab joondamiseks eestikeelse kõnetuvastusprojekti [26] käigus valminud akustilisi mudeleid ja samas projektis kasutatud häälduse sõnaraamatut[25].

Hääldus sõnaraamatu eesmärk on kirjapildis olevast sõnast teha sõna hääldus pärane vorm, kus kirjapildis olev sõna koosneb grafeemidest ja hääldus pärane vorm koosneb foneemidest [27]. A. Alumäe [28] on oma eestikeelse häälduse sõnaraamatus konverteerimise protsessi näiteks toonud, et kirjapildis oleva sõna „park“ hääldus pärane vorm on „p a r kk“, kus on märgata foneemi „kk“, näidates kirjapildi ja häälduse erinevust. A. Alumäe [28, 29] kasutab loodud sõnaraamatus reeglipõhist lähenemist, sest eesti keele kirjapilt ja hääldus on piisavalt sarnased, kuid võõrkeelsed sõnad ja nimed tuleb enne eraldi eestipärasemaks muuta. Näiteks on sõnaraamatu reeglistikus[26] inglisekeelse kirjapildi „Facebook“ eestikeelne vorm „feissbuk“.

M. Bacchiani [30] järgi on akustilise mudeli eesmärk *seada heli vastavusse mingile sümbolite jadale* ning selliseks sümbolite jadaks võib kasutada foneemide jada. Ta täpsustas, et

[23] https://bark.phon.ioc.ee/autosegment2/
[24] https://www.fon.hum.uva.nl/praat/manual/TextGrid.html
[25] https://github.com/alumae/et-g2p
[26] https://github.com/alumae/et-g2p/blob/master/src/etc/rules.yaml

akustiline mudel ennustab, mis on võimalike foneemide esinemise tõenäosus antud helis. Akustilisele mudelile antakse tavaliselt sisendiks helisignaali enda asemel heli iseloomustav raamitud tunnus, näiteks mel-sageduse kepstraal kordajad (ingl *Mel-frequency cepstral coefficients* ehk MFCC)[31], mis saadakse mel-spektrogrammist selle magnituudide ruutudest logaritmi võttes ja diskreetset koosinuse pööret rakendades [32]. Autosegmenteerijas kasutatud akustiline mudel [26] on närvivõrgupõhine ning selle treenimiseks on kasutatud palju tehisliku müraga rikastatud andmeid, seega suudab akustiline mudel ka halbades tingimustes helist foneeme ära tunda.

Autosegmenteerija kombineerib akustilise mudeli ja häälduse sõnaraamatu, et leida, mis ajahetkel helifailis iga foneem esineb. Kuna autosegmenteerija enda kohta puudub selgitav artikkel, on järgnevalt tegemist oletustega, mis põhinevad autosegmenteerija koodil[27] ja J. Yuani jt [33] joondamise kirjeldusel. Kuna akustiline mudel seab heli vastavusse foneemidega, siis autosegmenteerijale antud transkriptsioon viiakse esmalt häälduspärasesse vormi. Samuti vajab akustiline mudel sisendina heli iseloomustavat raamitud tunnust, mis autosegmenteerija puhul on MFCC väärtused, seega tuleb arvutada helifaili MFCC väärtused iga raami kohta. Järgnevalt luuakse Markovi peitmudel, mille seisundite jada pikkuseks on raamide arv ning seisunditeks on kõik häälduse sõnaraamatu foneemid, kuhu on lisatud juurde erilised vaikust ning tundmatut heli tähistavad foneemid. Seisundite (ehk antud juhul foneemide) üleminekutõenäosused on akustiline mudel õppinud varem ning vaatluse tõenäosuseks kasutatakse akustilise mudeli ennustust, kui suure tõenäosusega antud raamis kindel foneem esineb. Seejärel leitakse Viterbi algoritmiga kõige tõenäolisem foneemide jada paigutus helifailis, millest saab tuletada foneemide vahemikud helifailis, sest foneemid on leitud ühtlaste vahedega raamide pealt. Kuna helifailis esinev foneemide jada on teada, ei sobi kõik foneemide jadad, vaid ainult need, mis langevad kokku transkriptsioonist loodud häälduspärase vormiga.

3.4.2 Autosegmenteerija rakendamine vaikuse eemaldamiseks

Kuna autosegmenteerija vajab helifaili ja vastavat transkriptsiooni ning tagastab joondamisest sisaldava faili, saab töö jagada etappidesse. Esiteks peaks transkriptsioone töötlema, et neid häälduspärasemaks muuta, mille järel saab autosegmenteerijale anda ette töödeldud transkriptsiooni koos helifailiga. Et protsessi kiirendada, võiks kasutada lõimesid, et autosegmenteerija töötleks mitut helifaili korraga. Viimaks peaks tagastatud joondamisega faile rakendama originaalsetele helifailidele, et eemaldada otspunktidest vaikust ja kärpida kõnesiseseid pause.

Avalikult kasutatav autosegmenteerija veebirakendus on hea meetod algseks katsetamiseks, kuid suures koguses helifailide joondamiseks tuleks autosegmenteerija käivitada oma serveris. Selle jaoks saab kasutada autosegmenteerija lähtekoodi, milles on ka Dockerfile[28], millega saab rakendust oma serveris virtualiseerimise abil käivitada. Kuigi töö jaoks vajalikud arvutused tehakse muidu HPC [34] Rocket arvutusklastris, otsustati autosegmenteerija käivitada teises serveris, kus kasutajal oli rohkem õigusi, mistõttu oli programmi kergem katsetada.

Kuigi autosegmenteerija suudab populaarsemad võõrsõnad ja lühendid eesti keelele omasemaks muuta, kehtib see ainult piiratud hulgal võõrsõnadel. Lisaks vajavad numbrid, kuupäevad ja veebiaadressid töötlust, sest autosegmenteerija ei oska numbrite ja sümbolite häälдust kontekstipõhiselt leida. Sarnaselt RMSE ja ZCR baasil meetodile, kasutatakse transkriptsioonide häälduspärasemaks muutmisel I. Heina ja L. Rätsepa teksti eeltöötlejat. Peale

[27] https://github.com/alumae/kaldi-align-server
[28] https://docs.docker.com/engine/reference/builder/

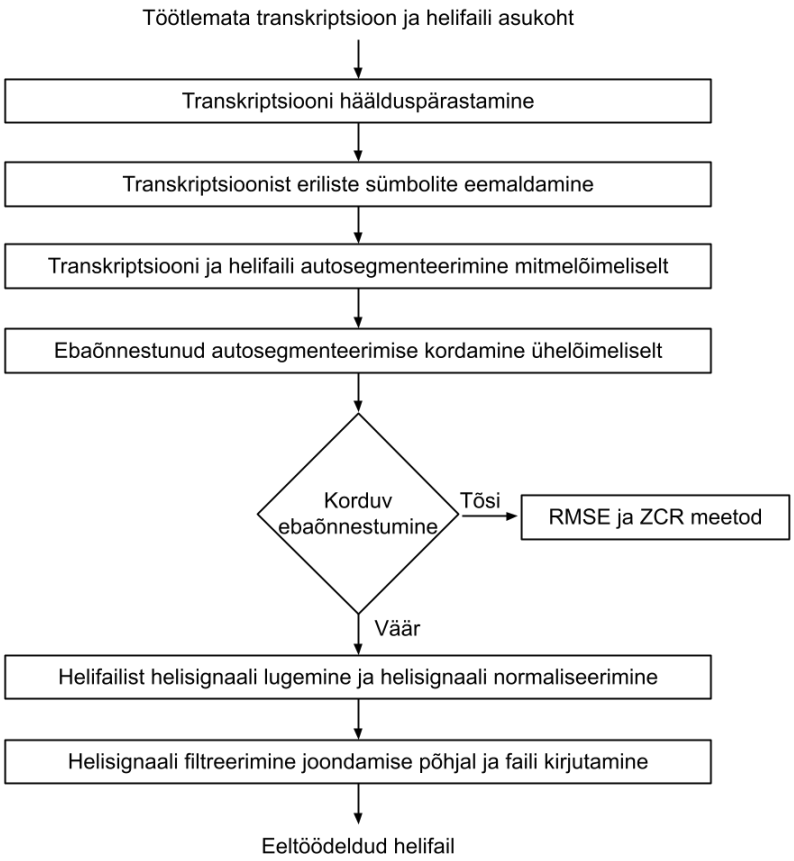
eeltöötlust eemaldatakse transkriptsioonist kõik sümbolid, mis pole tähed ega numbrid, sest autosegmenteerija ei joonda erilisi sümboleid sisaldavaid tekste.

Pärast transkriptsioonide töötlust rakendatakse autosegmenteerijat, kuid lõimede kasutamiseks peab enne autosegmenteerijat modifitseerima. Programmi kasutades selgus, et mitme helifaili samaaegsel joondamisel võib autosegmenteerija mõne helifaili joondamisel ebaõnnestuda ning ebaõnnestumise korral ei kustutata kõiki faile, mida joondamise ajal tekitati. Selliste failide suurus võis ulatuda kümnete megabaitideni ning tuhandete helifailide töötlusel võib serveris ruum kiiresti otsa saada. Probleemi lahenduseks muudeti autosegmenteerija koodi, et ebaõnnestumisel kõik loodud failid kustutatakse. Et ebaõnnestunud helifailid saaksid joondatud, rakendatakse autosegmenteerijat ebaõnnestunud helifailidele uuesti, seekord ühelõimeliselt.

Kuigi joondamisega helifailid on juba loodud, tehakse veel paar sammu enne helifailist vaikuse eemaldamist. Esmalt kontrollitakse, kas helifailile on ikka loodud joondamist sisaldav fail, sest harvadel juhtudel (0.1%) ei suuda autosegmenteerija helifaile ka ühelõimeliselt joondada. Sellistel juhtudel kasutatakse helifaili eeltöötlemiseks RMSE ja ZCR baasil meetodit. Kui joondamine õnnestus, loetakse helifailist sisse helisignaal, mis normaliseeritakse, et lõplikud helifailid oleksid sama valjud. Et joondust sisaldavast *TextGrid* tüüpi failist saaks joonduse informatsiooni sisse lugeda, kasutatakse Pythoni textgrid[29] moodulit.

Vaikusest puhastatud helisignaali loomiseks tehakse uus filtreeritud signaal, kuhu salvestakse ainult kõne ajal esinevad diskreedid. Kuna autosegmenteerija joondab helifailis muuhulgas vaikseid segmente, saab otspunktide vaikust eemaldada, kui jätta esimene ja viimane vaikuse segment filtreeritud signaali lisamata. Kuna helifail võib alata kõnega, näiteks Külli puhul, eemaldatakse helisignaalist esimene ja viimane vaikuse segment ainult siis, kui tegemist on täiesti otspunktides asuvate segmentidega. Kõnesiseste pauside kärpimiseks võrreldakse keskmiste vaikuse segmentide kestvuseid maksimaalse pausi kestvusega. Kui vaikus ületab maksimaalset kestvust, lisatakse filtreeritud signaali segmendi äärtest diskreete, kuni maksimaalne kestvus on saavutatud. Viimaks kirjutatakse filtreeritud helisignaal faili.

[29] http://github.com/kylebgorman/textgrid/



Joonis 16. Autosegmenteerija baasil eeltöötluse etapid.

Joonisel 16 on kokkuvõtteks kujutatud protsessi sammud, kuidas autosegmenteerija baasil meetod helifaile töötleb. Sarnaselt RMSE ja ZCR baasil meetodile, on tulemuseks normaliseeritud ning eemaldatud vaikusega helifail, mida saab edasi kasutada kõnesünteesi mudelite treenimiseks.

4. Treenimine ja tulemused

Loodud meetodite võrdlemiseks hinnatakse erinevalt eeltöödeldud helifaile. Kuna ainult eeltöödeldud helifailide põhjal ei saa analüüsida eeltöötluse mõju kõnesünteesile, treenitakse eeltöödeldud kõnekorpustel kõnesünteesi mudeleid ning hinnatakse loodud mudeleid. Järgnevalt on kirjeldatud kõnesünteesi mudelite loomise protsessi ning nii eeltöödeldud helifailide kui ka mudelite hindamiste tulemusi koos analüüsiga.

4.1 Kõnesünteesi mudelite loomine

Mudelite loomiseks on vaja kõnekorpuseid, mille peal mudeleid treenida. Seega mudelite loomise esimeseks sammuks on kõnekorpuste eeltöötlus. Töö raames eeltöödeldi täielikult nii UT uudiste kõnekorpus kui ka EKI ilukirjanduse kõnekorpus. Kuna ERR raadiouudiste kõnekorpust ei kasutata mudelite treenimisel, on aja kokkuhoiu eesmärgil ERR korpusest eeltöödeldud vaid osa faile. Eeltöödeldud korpused pole avalikustatud, kuid näitefaile on võimalik laadida alla loodud programmi repositooriumist[13]. Repositooriumis on leitavad ka loodud mudelid.

Töö raames on loodud kolm kõnesünteesi mudelit. Neist üks kasutas treenimiseks RMSE ja ZCR baasil eeltöödeldud kõnekorpuseid ja kaks autosegmenteerija baasil eeltöödeldud kõnekorpuseid. Autosegmenteerijat rakendavaid mudelid on loodud mitu, et kontrollida, kas transkriptsioonide eeltöötlemine enne autosegmenteerimist mõjutab tulemusi. Kuna autosegmenteerija eelduspäraseks sisendiks on transkriptsiooni häälduspärane kirjapilt, võiks transkriptsioonide eeltöötlus parandada autosegmenteerija tulemusi ja seeläbi ka eeltöödeldud kõnekorpustel treenitud mudelite tulemusi. Et seda oletust kontrollida, treenitakse kolm mudelit järgnevate nimedega: RMSE ja ZCR, autosegmenteerija eeltöötluseta ning autosegmenteerija eeltöötlusega.

Et hiljem oleks võimalik loodud mudeleid võrrelda kõnesünteesi projektis [4] varasemalt loodud mudelite, on mudelite loomisprotsess sarnane. Seega on kõnesünteesi mudelite loomiseks kasutatud eesti keelele kohandatud Deep Voice 3 tarkvara. Mudeleid treeniti TÜ teadusarvutuste keskuse [34] NVIDIA Tesla V100 GPU-d kasutades 32 päeva, mis erineb varasemate mudelite 45 päevast, kuid tulemused peaksid olema siiski võrreldavad. Sarnaselt varasematele mudelite, kasutati treenimiseks EKI ilukirjanduse kõnekorpust ja UT uudiste kõnekorpust, seega on loodud mudelid kuuele lugejale (Meelisele, Küllile, Marile, Albertile, Kalevile ja Vestale).

4.2 Kõnekorpuste eeltöötluse tulemused ja analüüs

Kuna kõnesünteesi mudelid võivad õppida vigu treenimiseks kasutatud korpustelt, hinnatakse esmalt eeltöötluse mõju kõnekorpustele endile. Nõnda saab mudelite hindamisel luua kergemini seoseid eeltöötlusega. Et kergendada seoste loomist rohkem, hinnatakse samu eeltöötluse meetodeid, mida kasutati mudelite loomisel. Kuna kõnekorpustes on kirjete arv suur ja hindamist peab viima läbi kolme meetodi jaoks, tehakse korpustest suvaline valim hinnatavaid helifaile. Valimi suuruseks on 200 helifaili nii EKI ilukirjanduse kõnekorpusest, UT uudiste kõnekorpusest kui ka ERR raadiouudiste kõnekorpusest, seega valitakse kokku 600 helifaili. Et tulemusi saaks täpsemalt võrrelda, jäetakse valitud helifailid samaks erinevate eeltöötluse meetodite puhul.

Hindamisel kuulatakse kõrvaklappidest hinnatavat eeltöödeldud helifaili, jälgitakse helifaili transkriptsiooni ja vaadatakse iga helifaili juures viit defekti:

1. Kas helifail algab järsult, näiteks jäetakse esimene häälik või sõna ütlemata.
2. Kas helifail lõppeb järsult, näiteks jäetakse viimane häälik või sõna ütlemata.

3. Kas helifaili alguses on vaikus. Peale helifailide, milles on märgatav paus helifaili alguse ja kõne alguse vahel, nimetatakse vaikuseks kõike peale inimkõne, sealhulgas mehhaanilisi helisid, hingeldusi ning suu häälitsusi.
4. Kas helifaili lõpus on vaikus. Sarnaselt eelmisele punktile, nimetatakse vaikuseks kõike peale inimkõne.
5. Kas helifaili keskel on kuulda sõnade katkemist, näiteks jääb lause keskel mõni sõna või häälik poolikuks.

Vaadeldavatest defektidest annavad esimesed neli informatsiooni otspunktides oleva vaikuse eemaldamise kvaliteedi kohta ning viies defekt hindab lausete keskel oleva vaikuse kärpimise kvaliteeti. Kuna hindamist viib läbi ainult töö autor, on kasutatud hindamismeetod subjektiivne, sest vigu antakse üksiku hindaja tunde järgi. Siiski võiks läbiviidud hindamine anda idee meetodite kvaliteedist ning võimaldada meetodeid omavahel võrrelda.

Tabelis 1 on toodud läbitud hindamise tulemused, mis on grupeeritud esmalt meetodi ja siis korpuse järgi. Lahtrites olevad väärtused näitavad, kui suurel osal helifailidest esinesid vaadeldud defektid. Pikem tabel, kus on näha hinnanguid igale kõnelejale eraldi, on toodud I lisas.

Tabel 1. Kõnekorpuste eeltöötluste meetodite hindamise tulemused.

Meetod ja korpus	Järsk algus	Järsk lõpp	Alguses vaikus	Lõpus vaikus	Keskel katkev
Autosegmenteerija eeltöötluseta	**1,3%**	**0,7%**	**6,0%**	**0,5%**	**0,3%**
EKI	0,5%	1,0%	0,5%	0,5%	0,0%
ERR	2,0%	0,5%	2,5%	0,5%	0,0%
UT	1,5%	0,5%	15,0%	0,5%	1,0%
Autosegmenteerija eeltöötlusega	**0,7%**	**0,7%**	**5,2%**	**0,7%**	**0,3%**
EKI	0,5%	1,0%	0,5%	0,5%	0,0%
ERR	1,5%	0,5%	2,0%	1,0%	0,0%
UT	0,0%	0,5%	13,0%	0,5%	1,0%
RMSE ja ZCR	**3,8%**	**2,5%**	**0,5%**	**19,0%**	**0,0%**
EKI	2,0%	6,0%	0,0%	1,0%	0,0%
ERR	8,0%	0,0%	0,0%	47,5%	0,0%
UT	1,5%	1,5%	1,5%	8,5%	0,0%

Vaadates meetodite tulemusi kõikide korpuste peale kokku, võib järeldada, et eeltöötluse rakendamine kõnekorpustele parandab kõnekorpuste kvaliteeti. Seda saab järeldada, sest eeltöötlusel tekkivad defektid (helifailide järsud lõpud ja algused ning katkevused helifailide keskel) on haruldased, samas on üldjuhul täidetud eeltöötluse eesmärk eemaldada vaikust otspunktidest. Kui eeltöödeldava kõnekorpuse kohta pole eelteadmisi, on soovitatav kasutada teksti eeltöötlusega autosegmenteerija meetodit, sest kasutatud meetodi puhul on tekkivate defektide tõenäosus väiksem kui 1% ning vaikuse säilimise tõenäosus alguses ja lõpus vastavalt 5,2% ja 0,7%. RMSE ja ZCR meetodi kasutamine pole nii täpne, sest tekkivate defektide arv on suurem ning täpsem vaikuse eemaldamine algusest ei kompenseeri ebatäpset lõpu töötlemist.

Kui teada eeltöödeldava korpuse tunnuseid, saab teha eeltöötluse meetodi valimisel parema otsuse, sest hinnatud meetodite tulemused erinevad sõltuvalt korpusest. ERR kõnekorpuse tulemuste põhjal on väidetav, et RMSE ja ZCR meetod ei oska eristada tugevaid hingetõmbeid kõnest, sest pooltel juhtudel jäid uudiste lugejate hingetõmbed helifaili sisse. Seda võivad põhjustada uudiste lugejate valjud ja pika kestvusega hingetõmbed, mille puhul on

RMSE ja ZCR väärtused kõrged. Kuna RMSE väärtus on pikka aega kõrge, eeldab meetod, et tegemist pole lühiajalise müraga ja jätab hingetõmbe helifaili. ZCR väärtust ei pruugi meetod isegi kasutada, sest ERR kõnekorpuse helifailid ei alga sageli pausiga, seega kasutab meetod staatilise lävega RMSE baasil eeltöötlust. Autosegmenteerijat kasutavate meetodite puhul esineb see defekt harva, kuna tõenäoliselt leiab autosegmenteerija teksti viimase hääliku asukoha ja eeldab, et edasi tuleb vaikus. Seega ka viimase hääliku järel olev hingetõmme klassifitseeritakse kui vaikus ja eemaldatakse.

ERR kõnekorpuse puhul ei suuda RMSE ja ZCR meetod ka algust kindlalt määrata, sest meetodiga loodud helifailides esineb järsk algus mitu korda rohkem võrreldes autosegmenteerijat kasutavate meetodiga. Vähendatud täpsuse võib põhjustada RMSE ja ZCR meetodi eeldus, et helifailis koheselt algav vali heli on ajutine mehhaaniline müra ning tuleb oodata esimest vaikset raami enne kõne alguse märkimist. Kuna ERR korpuse helifailid alustavad tihti kõnega ja kõnelejad räägivad kiiresti ja pausideta, võib meetod helifaili esimesi sõnu käsitleda kui müra ning neid eemaldada. Kuna autosegmenteerija ei kasuta helifaili alguses olevat vaikust kõne tuvastamisel, sest müraga arvestamine on akustilistesse mudelitesse sisse treenitud, ei esine autosegmenteerijat kasutavatel mudelitel sama probleemi. Et parandada RMSE ja ZCR meetodit, võiks määrata alguses olevale mürale lühikese maksimaalse kestvuse ning käsitleda kestvust ületanud heli kui kõne, mitte müra.

Sarnaselt ERR raadiouudiste kõnekorpusele, jätab RMSE ja ZCR meetod UT uudiste kõnekorpuse helifailide lõppu liigset vaikust. Vea sagedus pole nii sage kui ERR puhul, kuid võrreldes autosegmenteerija baasil meetoditega on erinevus märgatav. UT kõnekorpuse juures on probleemi põhjustajaks hiireklikk, mille eemaldamise tõenäosus sõltub kõnelejast. Kõige rohkem esineb viga Mari helifailides, kes ei jätnud kõne ja klikkide vahele pikka pausi ning kelle klikid on valjemad. Küllap on hiireklikid piisavalt valjud ja pikad, et RMSE märgendab neid kõnena, või nende ZCR väärtus on piisavalt suur, et kõne lõpp pikendatakse klikkideni. Sarnaselt hingetõmmetele esineb viga autosegmenteerijat kasutavate meetodite puhul harva.

Kuigi teised defektid esinevad autosegmenteerijat kasutavate meetodide puhul harva, ei suuda need meetodid eemaldada helifailide alguses olevat vaikust nii hästi kui RMSE ja ZCR meetod, mis on eriti märgatav UT kõnekorpuse puhul. UT kõnekorpuste alguses esinev klikilik mehhaaniline müra on kuulda mitmetes autosegmenteerijaga eeltöödeldud helifailides. Vaadates tulemusi kõneleja põhiselt, esineb defekt kõige rohkem Mari helifailides ning väiksemal määral Alberti helifailides. Teiste kõnelejate puhul pole defekti märgata. Defekti põhjuseks võib olla Mari ja Alberti klikkide valjus ja sarnasus nende häältega, seega määrab autosegmenteerija igaks juhuks, et tegu pole vaikusega vaid tundmatu heliga. Kuna autosegmenteerija võib tundmatut heli märkida ka lühendite ja kokutamiste juures, võib tundmatu heli kustutamine tekitada defekte lausete keskel. Eelistatumalt võib proovida täpsuse parandamiseks kustutada tundmatut heli vaid lause alguses, enne esimese sõna ära tundmist.

Eeltöötlusega ja eeltöötluseta autosegmenteerijat kasutavate meetodite puhul on omavaheline erinevus märgatav UT kõnekorpuse defektide korral, kus teksti eeltöötlus vähendab helifailides alguses olevaid defekte. Tekstile eeltöötluse rakendamine on tõenäoliselt märgatav ainult UT kõnekorpuse juures, sest selle transkriptsioon sisaldab teiste korpustega võrreldes palju rohkem lühendeid ja numbreid. Täpsus võib paraneda, sest autosegmenteerija sisendiks on tihedamini esimese sõna häälduspärane kirjapilt, mis võib aidata esimese sõna asukoha määramisel, sest autosegmenteerija kasutab häälikupõhist mudelit. Kuna esimese sõna asukoht on kindlamini määratud, väheneb tõenäosus märgendada esimesele sõnale eelnevat müra tundmatu helina või märgendada esimesi häälikuid vaikusena.

Tulemuste kokkuvõtteks võib öelda, et kui eeltöödeldava kõnekorpuse helifailid algavad piisavalt pika pausiga, võib kasutada RMSE ja ZCR meetodit ka mehhaanilise müra olemasolul ning saada väheste defektidega parandatud kõnekorpus. Kuigi autosegmenteerijat kasutava meetodiga saaks helifailide lõpus olevat vaikust täpsemini eemaldada, võib autosegmenteerija helifailide algust ebatäpsemalt töödelda. Lisaks on RMSE ja ZCR meetodi kasutamine kiirem, sest see ei vaja autosegmenteerija jooksutamist. Siiski annab autosegmenteerija üldjuhul parema tulemuse, eriti kui helifailide alguses puudub paus, sest sellised helifailid ei täida RMSE ja ZCR meetodi eeldusi. Kui korpuse transkriptsioonid pole häälduspäraselt kirjutatud, annab teksti eeltöötlusega autosegmenteerija meetod parima tulemuse, samas väheste numbrite ja lühenditega transkriptsiooni puhul pole teksti eeltöötlus vajalik.

4.3 Kõnesünteesi mudelite tulemused ja analüüs

Kuna loodud kõnesünteesi mudeleid hinnatakse sünteesitud helifailide kaudu, peavad mudelid esiteks helifaile sünteesima. Sünteesitud helifailide tekstid saadakse kõnekorpuste testhulgast, mis eraldati enne mudelite treenimist. Kuna EKI kõnelejaid on vähem, sünteesitakse nii Meelisele kui ka Küllile 100 helifaili iga mudeli kohta ning igale UT kõnelejale 62 faili. Sellise hindamise puhul kuulatakse iga mudeli juures 448 faili, mis võiks anda võrreldavad tulemused. Sarnaselt eeltöötluse hindamisele, jäetakse hinnatavad tekstid mudelite vahel samaks, et oleks võimalik võrrelda hindeid spetsiifiliste helifailide juures.

Et tulemusi võrrelda varasemate mudelitega [4], hinnatakse samu defekte, mida hinnati eelmiste mudelite puhul. Nendest defektidest tehakse valik, kuna eeltöötlus ei pruugi kõiki kategooriad mõjutada ning nii pole hindaja tähelepanu paljude defektide vahel hajutatud. Hindamisel kuulatakse kõrvaklappidest hinnatavat sünteesitud helifaili ja vaadatakse iga helifaili juures kolme defekti:

1. Kas helifailis hääldatakse lühendeid, numbreid ja sümboleid valesti või jäetakse need vahele. Veaks loetakse ka juhte, kus helifailis on kuulamist segavaid probleeme lühendile, numbrile või sümbolile koheselt eelneva või järgneva sõna juures.
2. Kas helifail algab või lõppeb järsult. Veaks loetakse juhte, kus transkriptsiooni algust või lõppu ei loeta üldse või helitugevus on kuulamist segavalt madal alguses või lõpus.
3. Kas helifailis esineb helitugevuse probleeme. Veaks loetakse juhte, kus kuulamisel on mingil hetkel helitugevus nii madal, et loetud tekstist on keskendumiseta raske või võimatu aru saada. Järsud helifaili algused ja lõpud ei loeta veaks, kui helifail ise algab või lõppeb poole sõna pealt, sest siis pole probleemiks helitugevus vaid heli puudumine.

Vaadeldavatest defektidest võiks esimene hinnata teksti eeltöötluse mõju autosegmenteerija meetoditele ning viimased kaks defekti võiksid hinnata vaikuse eemaldamise mõju kõikidele meetoditele. Sealjuures teine defekt võiks täpsemalt hinnata otspunktide vaikuse mõju ja kolmas defekt lausete keskel oleva vaikuse mõju. Sarnaselt eeltöötluse hindamisele viib hindamist läbi töö autor, seega võrdlus varasemate mudelite ei pruugi olla täpne, kuid loodud meetodite omavaheline võrdlus võiks siiski olla realistlik.

Tabelis 2 on toodud läbitud hindamise tulemused, mis on grupeeritud esmalt mudeli ja seejärel kõneleja järgi. Lahtrites olevad väärtused näitavad, kui suurel osal helifailidest esinevad vaadeldud defektid.

Tabel 2. Kõnesünteesi mudelite hindamise tulemused.

Mudel ja lugeja	Numbrite, sümbolite, lühendite vead	Järsk lõpp/algus	Helitugevuse probleemid
Autosegmenteerija eeltöötluseta	**1,8%**	**2,9%**	**6,9%**
Albert	3,2%	4,8%	9,7%
Kalev	4,8%	0,0%	6,5%
Külli	0,0%	4,0%	3,0%
Mari	3,2%	4,8%	9,7%
Meelis	0,0%	2,0%	4,0%
Vesta	1,6%	1,6%	12,9%
Autosegmenteerija eeltöötlusega	**1,6%**	**3,3%**	**6,3%**
Albert	3,2%	8,1%	9,7%
Kalev	3,2%	3,2%	8,1%
Külli	0,0%	2,0%	1,0%
Mari	3,2%	3,2%	9,7%
Meelis	0,0%	2,0%	3,0%
Vesta	1,6%	3,2%	11,3%
RMSE ja ZCR	**1,6%**	**5,1%**	**6,9%**
Albert	3,2%	8,1%	8,1%
Kalev	3,2%	1,6%	4,8%
Külli	0,0%	5,0%	3,0%
Mari	3,2%	4,8%	11,3%
Meelis	0,0%	5,0%	7,0%
Vesta	1,6%	6,5%	9,7%
Varasemad mudelid	**3,6%**	**15,9%**	**17,0%**
Albert	1,7%	20,0%	14,8%
Kalev	3,4%	13,8%	10,3%
Külli	3,5%	6,2%	19,5%
Mari	4,3%	20,9%	29,3%
Meelis	4,2%	9,2%	15,8%
Vesta	4,4%	25,4%	12,3%

Märkus: kuna L. Rätsepa jt [4] loodud raportis varasemate kõnesünteesi mudelite kohta on toodud vigade tõenäosus ainult kõneleja järgi, on kasutatud kõigi kõnelejate vigade tõenäosuse leidmiseks aritmeetilist keskmist. See ei pruugi olla täiesti täpne, sest erinevate kõnelejate kohta hinnati projektis erinev arv lauseid.

Tabeli põhjal saab väita, et eeltöötluse rakendamisel on positiivne mõju treenitud mudelitele, sest iga vaadeldud defekti arv kahanes võrreldes varasemate mudelitega. Samas on tõenäoline, et eeltöötlus ei mõjuta tegelikult numbrite, sümbolite ja lühendite vigu, sest ka varasemate mudelite puhul rakendati transkriptsioonidele sama eeltöötlust. Vähenenud vigade arvu võib põhjustada näiteks hindajate erinev standard. Samuti on helitugevuse probleemide vähenemine kaheldav, sest eeltöötluse tulemuste alusel peaksid autosegmenteerijat kasutavad meetodid RMSE ja ZCR meetodiga võrreldes vaikust paremini eemaldama. Aga see meetodite kvaliteedi vahe ei kajastu mudelite hindamise tulemustes, sest kõikide loodud mudelite tulemused on sarnased. Suurima tõenäosusega põhjustab vähenenud vigade arvu samuti hindajate erinev standard. Viimase defekti, järsu lõpu ja alguse kohta võib kõige kindlamini väita, et eeltöötlus vähendab vea esinemist. See on väidetav, sest võrreldes teiste

defektidega, on selle vea hindamine objektiivsem ning loodud meetodite kvaliteedi erinevus kajastub mudelite tulemuses.

Võrreldes kolme loodud mudelit on numbrite, sümbolite ja lühendite vigade arv sarnane. Seega võib oletada, et teksti eeltöötlus võib parandada autosegmenteerija tulemusi eeltöötlusel, kuid mitte piisaval määral, et parandused kajastuksid treenitud mudelites. Tulemusi võivad mõjutada hinnatud transkriptsioonid, kus esines nii veebiaadress kui ka ebatavalisi sümbolite kasutusi, näiteks „Kalev/Cramo“, mille häädusi ei pruugi mudel isegi perfektselt eeltöödeldud kõnekorpuse puhul ära õppida, seega tõuseb vigade arv sõltumata meetodist.

Autosegmenteerijat kasutavad mudelid teevad vähem järsu lõpu ja alguse vigu, mis on kooskõlas eeltöötluse tulemustega, kus autosegmenteerija meetodite korral esineb vähem järske lõppe ja alguseid. Võrreldes korpuste asemel kõnelejaid, on nii mudelite kui ka eeltöötluse puhul EKI kõnelejate juures vigade erinevus suurim. Järelikult saab väita, et eeltöödeldud kõnekorpuste järsud otspunktid kanduvad üle treenitud mudelitesse.

Kuigi tulemustesse seda ei märgitud, märgati hindamisel autosegmenteerija mudelite puhul helifailide alguses mehhaanilist klikki. Klikki sisaldasid üksikud helifailid, mis esinesid vaid Alberti ja Mari puhul. ZCR ja RMSE mudelil seda defekti ei märgatud. Tõenäoliselt õppisid autosegmenteerija mudelid ära kõnekorpustes esineva mehhaanilise kliki, mida eeltöötlusel ei suutnud autosegmenteerija meetod Alberti ja eriti Mari puhul eemaldada.

Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli luua programm, mis eemaldaks erinevate meetoditega eestikeelsete kõnekorpuste helifailidest vaikust. Meetodeid rakendati mitmele kõnekorpusele, mida kasutati kõnesünteesi mudelite loomiseks. Nii eeltöödeldud kõnekorpuseid kui ka kõnesünteesi mudeleid hinnati ja analüüsiti, et määrata meetodite kvaliteeti.

Töö raames analüüsiti kolme kõnekorpust ja leiti neis esinevad defektid ning iseloomulikud omadused. EKI ilukirjanduse kõnekorpust ja UT uudiste kõnekorpust kasutati meetodite arendamiseks ja mudelite loomiseks ning ERR raadiouudiste kõnekorpust kasutati meetodite kontrollimiseks väljaspool arenduskeskkonda.

Esmalt loodi eeltöötluse jaoks RMSE ja ZCR baasil meetod, mis kasutas kahekihilist RMSE leidmist, et leebemalt kärpida vaikust helifailide keskelt. Loodud meetod kasutab dünaamilist läve, kui helifail algab vaikusega, kuid vaikuse puudumisel kasutatakse staatilist läve. Meetodile on juurde lisatud elemente, nagu kõne miinimumkestvus ning transkriptsiooniga arvestamine, et tulemusi vastavalt kõnekorpuses leitud defektidele parandada.

Teisena loodi autosegmenteerija baasil meetod, mis rakendab varasemalt loodud programmi transkriptsiooni ja helifaili joondamiseks ning eemaldab leitud vaikused. Et autosegmenteerijat saaks kasutada lõimedega, muudeti algset programmi ning ilmekate pauside säilitamiseks töödeldi autosegmenteerija tulemusi.

Kokku treeniti Deep Voice 3 tarkvaral kolm kõnesünteesi mudelit, mis kasutasid loodud programmiga eeltöödeldud kõnekorpuseid. Nii treenimiseks kasutatud eeltöödeldud kõnekorpuseid kui ka loodud mudeleid hinnati ja võrreldi. Kõnesünteesi mudeleid võrreldi lisaks varasema kõnesünteesi projekti mudelitega.

Eeltöötluse hindamisel kinnitati, et nii autosegmenteerija kui ka RMSE ja ZCR baasil meetod parandavad kõnekorpuste kvaliteeti, kuid autosegmenteerija sõltub vähem kõnekorpuste omadustest ja on täpsem. Lisaks järeldati mudelite hindamisel, et eeltöödeldud kõnekorpustel treenitud mudelid teevad vähem järsu alguse ja lõpu vigu.

Kuna hindamisel märgati mõlema meetodi puhul vigu, mille parandamine tundub võimalik, saaks loodud meetodeid edasi arendada ning hindamist korrata. Uute meetodite lisamine programmi oleks samuti huvitav, näiteks võiks kasutada närvivõrgupõhist lähenemist. Sellise lähenemise jaoks oleks vaja märgendada käsitsi kõnekorpuste helifailides olevate vaikuste asukohad, et helifaile saaks närvivõrgu sisendina kasutada, mistõttu töös seda meetodit ei kasutatud.

Peale meetodite parandamise ja loomise, oleks täpsemate tulemuste ja kindlamate oletuste saamiseks vajalik objektiivsem hindamine. Üheks võimaluseks oleks viia hindamist läbi suure grupiga, et tulemusi ühtlustada. Kuna helifailide kuulamine on pikk protsess, on sellise grupi leidmine keeruline, mistõttu töös seda võimalust ei kasutatud. Samuti saaks tulemusi rohkem täpsustada, kui kasutada mitmeid kõnesünteesi tarkvarasid. Siis saaks kindlamalt väita, et leitud tulemused pole kasutatud tarkvarast sõltuvad.

Viidatud kirjandus

[1] L. R. Rabiner; M. R. Sambur. An algorithm for determining the endpoints of isolated utterances. The Bell System Technical Journal, vol. 54, no. 2, 1975, 297–315. https://ieeexplore.ieee.org/document/6778857 (01.04.2021)

[2] L.-S. Huang; C.-H. Yang. A novel approach to robust speech endpoint detection in car environments. 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100), Istanbul, Turkey, 2000, 1751–1754. https://ieeexplore-ieee-org.ezproxy.utlib.ut.ee/document/862091 (01.04.2021)

[3] B. Chandu; A. Munikoti; K. S. Murthy; G. Murthy; V. C. Nagaraj. Automated Bird Species Identification using Audio Signal Processing and Neural Networks. International Conference on Artificial Intelligence and Signal Processing (AISP), Amaravati, India, 2020, 1–3. https://ieeexplore-ieee-org.ezproxy.utlib.ut.ee/document/9073584 (29.03.2021)

[4] L. Rätsep; L. Piits; H. Pajupuu; I. Hein; M. Fišel. Närvivõrgu põhise kõnesünteesi arendamine. Tehniline raport. Tartu Ülikool, arvutiteaduse instituut, Eesti Keele Instituut, 06.10.2020, 1–7. https://arxiv.org/pdf/2010.02636.pdf (08.12.2020)

[5] G. S. Ying; C. D. Mitchell; L. H. Jamieson. Endpoint detection of isolated utterances based on a modified Teager energy measurement. 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, Minneapolis, MN, USA, 1993, 732–733 vol.2. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=319416 (29.03.2021)

[6] M. Fišel; L. Rätsep. Eesti uudislausete kõnekorpus. 2020. https://doi.org/10.15155/9-00-0000-0000-0000-001ABL (31.03.2021)

[7] National Instruments. Understanding FFTs and Windowing. Instrument Fundamentals: Complete Guide, 8–11. https://download.ni.com/evaluation/pxi/Understanding%20FFTs%20and%20Windowing.pdf (12.04.2021)

[8] numpy.hanning. NumPy Documentation. https://numpy.org/doc/stable/reference/generated/numpy.hanning.html (12.04.2021)

[9] V. Velardo. Short-Time Fourier Transform Explained Easily. https://www.youtube.com/watch?v=-Yxj3yfvY-4 (12.04.2021)

[10] V. Velardo. How to Extract Audio Features. https://www.youtube.com/watch?v=8A-W1xk7qs8 (26.04.2021)

[11] J. O. Smith. Spectrograms. Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications, Second Edition, Online book, 2007 edition. https://ccrma.stanford.edu/~jos/st/Spectrograms.html (23.04.2021)

[12] V. Velardo. Mel Spectrograms Explained Easily. https://www.youtube.com/watch?v=9GHCiiDLHQ4 (12.04.2021)

[13] W. Ping; K. Peng; A. Gibiansky; S. O. Arik; A. Kannan; S. Narang; J. Raiman; J. Miller. Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning. ICLR 2018, 1–9. https://arxiv.org/abs/1710.07654v3 (11.04.2021)

[14] T. H. Zaw; N. War. The combination of spectral entropy, zero crossing rate, short time energy and linear prediction error for voice activity detection. 2017 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 2017, 1–2. https://ieeexplore.ieee.org/document/8281794 (01.04.2021)

[15] M. Jalil; F. A. Butt; A. Malik. Short-time energy, magnitude, zero crossing rate and autocorrelation measurement for discriminating voiced and unvoiced segments of speech signals. 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), Konya, Turkey, 2013, 208–209. https://ieeexplore.ieee.org/document/6557272 (31.03.2021)

[16] A. Ganapathiraju; L. Webster; J. Trimble; K. Bush; P. Kornman. Comparison of energy-based endpoint detectors for speech signal processing. Proceedings of SOUTHEASTCON '96, Tampa, FL, USA, 1996, 500–503. https://ieeexplore-ieee-org.ezproxy.utlib.ut.ee/document/510121 (01.04.2021)

[17] V. Panayotov; G. Chen; D. Povey; S. Khudanpur. Librispeech: An ASR corpus based on public domain audio books. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, 5206–5210. https://www.danielpovey.com/files/2015_icassp_librispeech.pdf (28.04.2021)

[18] K. Park; T. Mulc. CSS10: A Collection of Single Speaker Speech Datasets for 10 Languages. Proc. Interspeech 2019, 2019, 1566–1570. https://arxiv.org/abs/1903.11269 (29.04.2021)

[19] E. K. Instituut. Ilukirjanduse kõnekorpused Meelis ja Külli. 2020.

[20] E. Meister. Corpus of Radio News. Center of Estonian Language Resources, 2014. https://doi.org/10.15155/9-00-0000-0000-0000-00021L (17.04.2021)

[21] B. McFee; V. Lostanlen; A. Metsai; M. McVicar; S. Balke; C. Thomé; C. Raffel; F. Zalkow; A. Malek; Dana; K. Lee; O. Nieto; J. Mason; D. Ellis; E. Battenberg; S. Seyfarth; R. Yamamoto; K. Choi; viktorandreevichmorozov; J. Moore; R. Bittner; S. Hidaka; Z. Wei; nullmightybofo; D. Hereñú; F.-R. Stöter; P. Friesch; A. Weiss; M. Vollrath; T. Kim. librosa/librosa: 0.8.0 (Version 0.8.0). Zenodo, 22.07.2020. http://doi.org/10.5281/zenodo.3955228 (18.04.2021)

[22] J. Shen; J. Hung; L. Lee. Robust entropy-based endpoint detection for speech recognition in noisy environments. ICSLP, 1998, 1–2. http://www.mirlab.org/jang/books/audioSignalProcessing/paper/endPointDetection/shenHL98-endpoint.pdf (19.04.2021)

[23] H. Läänemets. Autoregressiivsed peidetud Markovi mudelid. Magistritöö, Tartu Ülikool, loodus- ja täppisteaduste valdkond, juhendaja: Märt Möls, 2017, 6–11. http://dspace.ut.ee/bitstream/handle/10062/57095/laanemets_hanna_msc_2017.pdf (30.04.2021)

[24] librosa. Viterby decoding. https://librosa.org/doc/main/auto_examples/plot_viterbi.html#sphx-glr-auto-examples-plot-viterbi-py (29.04.2021)

[25] G. Anbarjafari. Aliasing and image enhancement. Digital Image Processing, 2014. https://sisu.ut.ee/imageprocessing/book/4 (01.05.2021)

[26] T. Alumäe; O. Tilk; Asadullah. Advanced Rich Transcription System for Estonian Speech. Frontiers in Artificial Intelligence and Applications, Volume 307: Human Language Technologies – The Baltic Perspective, 1–8. https://doi.org/10.3233/978-1-61499-912-6-1 (02.05.2021)

[27] S. Yolchuyeva; G. Németh; B. Gyires-Tóth. Grapheme-to-Phoneme Conversion with Convolutional Neural Networks. Applied Sciences, 2019, 9, 1143, 1–2. https://doi.org/10.3390/app9061143 (02.05.2021)

[28] T. Alumäe. README.md. et-g2p. https://github.com/alumae/et-g2p (02.05.2021)

[29] T. Alumäe. Recent improvements in Estonian LVCSR. Spoken Language Technologies for Under-Resourced Languages, 2014. https://www.researchgate.net/publication/261610480_Recent_improvements_in_Estonian_LVCSR (02.05.2021)

[30] M. Bacchiani. Google Research on End-to-End Models for Speech Recognition -English version- . https://www.youtube.com/watch?v=LTOu9_IWMyQ (02.05.2021)

[31] V. Peddinti; D. Povey; S. Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. INTERSPEECH-2015, 2015, 3214–3218. https://www.danielpovey.com/files/2015_interspeech_multisplice.pdf (03.05.2021)

[32] V. Velardo. Mel-Frequency Cepstral Coefficients Explained Easily. https://www.youtube.com/watch?v=4_SH2nfbQZ8 (03.05.2021)

[33] J. Yuan; W. Lai; C. Cieri; M. Liberman. Using Forced Alignment for Phonetics Research. 2018, 1–2. http://languagelog.ldc.upenn.edu/myl/ForcedAlignment_Final_edited.pdf (03.05.2021)

[34] University of Tartu. *UT Rocket.* share.neic.no. https://doi.org/10.23673/PH6N-0144 (23.04.2021)

Lisad

I. Heli eeltöötluse tulemuste analüüsi pikendatud tabel

Meetod ja lugeja	Järsk algus	Järsk lõpp	Alguses vaikus	Lõpus vaikus	Keskel katkev
Autosegmenteerija eeltöötluseta	**1,3%**	**0,7%**	**6,0%**	**0,5%**	**0,3%**
EKI	**0,5%**	**1,0%**	**0,5%**	**0,5%**	**0,0%**
Külli	1,1%	2,2%	0,0%	0,0%	0,0%
Meelis	0,0%	0,0%	0,9%	0,9%	0,0%
ERR	**2,0%**	**0,5%**	**2,5%**	**0,5%**	**0,0%**
Birgit_Itse	3,1%	0,0%	3,1%	0,0%	0,0%
Indrek_Kiisler	3,4%	0,0%	0,0%	0,0%	0,0%
Kai_Vare	6,9%	0,0%	3,4%	0,0%	0,0%
Meelis_Kompus	0,0%	0,0%	3,0%	0,0%	0,0%
Tarmo_Maiberg	0,0%	0,0%	0,0%	0,0%	0,0%
Tõnu_Karjatse	0,0%	0,0%	0,0%	0,0%	0,0%
Vallo_Kelmsaar	0,0%	3,3%	6,7%	3,3%	0,0%
UT	**1,5%**	**0,5%**	**15,0%**	**0,5%**	**1,0%**
Albert	0,0%	0,0%	22,2%	0,0%	0,0%
Kalev	5,5%	0,0%	0,0%	0,0%	0,0%
Mari	0,0%	0,0%	41,7%	2,1%	4,2%
Vesta	0,0%	1,9%	0,0%	0,0%	0,0%
Autosegmenteerija eeltöötlusega	**0,7%**	**0,7%**	**5,2%**	**0,7%**	**0,3%**
EKI	**0,5%**	**1,0%**	**0,5%**	**0,5%**	**0,0%**
Külli	1,1%	2,2%	0,0%	0,0%	0,0%
Meelis	0,0%	0,0%	0,9%	0,9%	0,0%
ERR	**1,5%**	**0,5%**	**2,0%**	**1,0%**	**0,0%**
Birgit_Itse	0,0%	0,0%	3,1%	0,0%	0,0%
Indrek_Kiisler	3,4%	0,0%	0,0%	0,0%	0,0%
Kai_Vare	6,9%	0,0%	0,0%	0,0%	0,0%
Meelis_Kompus	0,0%	0,0%	6,1%	0,0%	0,0%
Tarmo_Maiberg	0,0%	0,0%	0,0%	0,0%	0,0%
Tõnu_Karjatse	0,0%	0,0%	0,0%	0,0%	0,0%
Vallo_Kelmsaar	0,0%	3,3%	3,3%	6,7%	0,0%
UT	**0,0%**	**0,5%**	**13,0%**	**0,5%**	**1,0%**
Albert	0,0%	0,0%	15,6%	0,0%	0,0%
Kalev	0,0%	0,0%	0,0%	0,0%	0,0%
Mari	0,0%	0,0%	39,6%	2,1%	4,2%
Vesta	0,0%	1,9%	0,0%	0,0%	0,0%
RMSE ja ZCR	**3,8%**	**2,5%**	**0,5%**	**19,0%**	**0,0%**
EKI	**2,0%**	**6,0%**	**0,0%**	**1,0%**	**0,0%**
Külli	3,2%	7,5%	0,0%	0,0%	0,0%
Meelis	0,9%	4,7%	0,0%	1,9%	0,0%
ERR	**8,0%**	**0,0%**	**0,0%**	**47,5%**	**0,0%**
Birgit_Itse	3,1%	0,0%	0,0%	46,9%	0,0%
Indrek_Kiisler	6,9%	0,0%	0,0%	69,0%	0,0%

Kai_Vare	6,9%	0,0%	0,0%	17,2%	0,0%
Meelis_Kompus	12,1%	0,0%	0,0%	42,4%	0,0%
Tarmo_Maiberg	3,6%	0,0%	0,0%	57,1%	0,0%
Tõnu_Karjatse	15,8%	0,0%	0,0%	52,6%	0,0%
Vallo_Kelmsaar	10,0%	0,0%	0,0%	50,0%	0,0%
UT	**1,5%**	**1,5%**	**1,5%**	**8,5%**	**0,0%**
Albert	2,2%	0,0%	2,2%	0,0%	0,0%
Kalev	1,8%	0,0%	3,6%	7,3%	0,0%
Mari	2,1%	2,1%	0,0%	14,6%	0,0%
Vesta	0,0%	3,8%	0,0%	11,5%	0,0%

II. Litsents

Mina, Andreas Teder,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

   Kõnekorpuste audiofailide automaatne eeltöötlus kõnesünteesi treenimiseks,

   mille juhendaja on Liisa Rätsep,

   reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commonsi litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.

3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.

4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Andreas Teder

07.05.2021